Commodore

COMPUTER CLUB

54

L. 4.500

La rivista degli utenti di sistemi Commodore

25 giugno 1988 - Anno VII - N. 54
Sped. Abb. Post. Gr. III/70 - CR

Distr. MePe

CAMPUS

- Gli errori di chi comincia
- Poke segrete del C 64
- Usiamo il macroassembler

TEST PANASONIC KX-P1081

BIT PARADE

GIOCHI DEL MESE

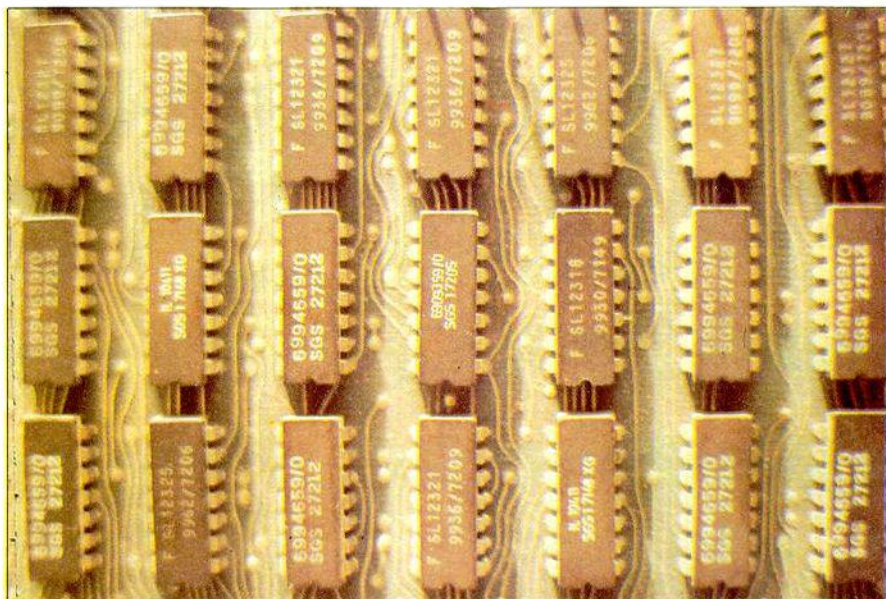
GRAFICA IN PROSPETTIVA

TUTTI IN FORMA COL C/64

Commodore COMPUTER CLUB

54 Sommario

RUBRICHE

Direttore: Alessandro de Simone - **Caporedattore:** Michele Maggi
Redazione/collaboratori: Paolo Agostini, Davide Ardizzone, Claudio Baiocchi, Luigi Callegari, Sergio Camici, Umberto Colapicchioni, Maurizio Dell'Abate, Valerio Ferri, Roberto Ferro, Cristina Magnaghi, Giancarlo Mariani, Roberto Marigo, Clizio Merli, Marco Mietta, Marco Miotti, Oscar Moccia, Roberto Morassi, Guido Pagani, Antonio Pastorelli, Sonja Scharrer, Fabio Sorgato, Valentino Spataro, Danilo Toma
Grafica: Arturo Ciaglia, Elena Salvadori
Direzione, redazione, pubblicità: v.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Milano: Leandro Nencioni (direttore vendite), Giorgio Ruffoni - Viale Famagosta, 75 - 20142 Milano Tel. 02/8467348
• Emilia Romagna: Spazio E - P.zza Roosevelt, 4 - 40123 Bologna - Tel. 051/236979
• Toscana, Marche, Umbria: Mercurio srl - via Rodari, 9 - San Giovanni Valdarno (Ar) - Tel. 055/947444
• Lazio, Campania: Spazio Nuovo - via P. Foscari, 70 - 00139 Roma - Tel. 06/8109679
Segreteria: Tiziana Sodano - **Abbonamenti:** Liliana Spina
Tariffe: prezzo per copia L. 4.500. Abbonamento annuo (11 fascicoli) L. 45.000. Estero: il doppio.
Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 85.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizione: Systems Editoriale Srl - **Fotolito:** Systems Editoriale Srl
Stampa: Systems Editoriale
Registrazioni: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70%
Distrib.: MePe - via G. Carcano, 32 - Milano
Periodici Systems: Banca Oggi - Commodore Club (disco) - Commodore Computer Club - Commodore Computer Club (disco produzione tedesca) - Computer - Computer disco - Electronic Mass Media Age - Energy Manager - Hospital Management - MondoRicambi - Nursing '90 - PC Programm (disco) - Personal Computer - Security - Software Club (cassetta ed. italiana) - VR Videoregistrare

LE SPECULAZIONI INFORMATICHE

Da un po' di tempo c'è penuria di Ram. A chi giova?

Il mondo occidentale si fonda sulla libera concorrenza, lo sappiamo tutti.

Molto spesso tale forma di mercato implica vantaggi per il consumatore perchè costringe i produttori ad offrire, a prezzi competitivi, prodotti sempre migliori ad un pubblico sempre più esigente.

A volte, però, possono crearsi condizioni "al contorno" piuttosto particolari in cui i produttori sembrano mettersi d'accordo su alcuni punti che impongono, al contrario, un atteggiamento di tipo monopolistico.

Un esempio, che ancora scotta, è fornito dal gioco di borsa che, se ben condotto, è indice della reale condizione di salute del mercato in cui opera. Altre volte, al contrario, può essere influenzato da giochi al ribasso o al rialzo, più o meno puliti, in cui, troppo spesso, a lasciare le penne sono soprattutto i piccoli risparmiatori.

Se una fabbrica produce poco, e non riesce a soddisfare le richieste, altre fabbriche incrementano la propria produzione; oppure nuovi stabilimenti, addirittura, vengono realizzati approfittando del momento favorevole.

Negli ultimi tempi, però, assistiamo ad un fenomeno davvero inconsueto: sia l'occidente (Stati Uniti in testa) che l'oriente (Taiwan per prima) sembra che non riescano a soddisfare le richieste di Ram (badate bene: solo di Ram) da parte dei produttori di computer.

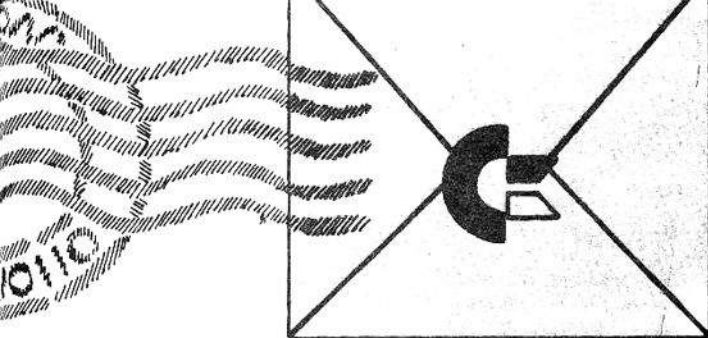
Semplice imprevidenza, "esplosione" inaspettata delle richieste? Eppure era noto da tempo che il progresso tecnologico avrebbe messo a disposizione calcolatori in grado di gestire quantità considerevoli di Ram, soprattutto tenendo conto del mercato relativo alla grafica computerizzata e al DTP (desk top publishing).

Sembra un po' strano, infatti, che tutti i produttori, nello stesso momento, avvertano i loro abituali clienti della penuria di Ram. Nelle circolari si annuncia il "ritocco" dei chip che, ormai, hanno un prezzo decisamente consistente: all'utente finale la Ram costa oggi circa 1000 lire a K/byte, vale a dire il triplo di quanto costava poco tempo fa.

Siccome le fabbriche di chip non sono pochissime, ma tutte in rigida concorrenza tra loro, non può sfuggire la singolarità del fenomeno che le vede, paradossalmente, concordi nell'annunciare la penuria di Ram e (ma solo come incresciosa, indiretta e involontaria conseguenza) l'aumento del loro prezzo.

E tutto questo mentre le fabbriche tendono a sfornare computer che richiedono enormi quantità di questi preziosi chip.

Che strana combinazione!

Alessandro de Simone

la vostra posta

DOVE SBAGLIO?

□ Il programma che ho scritto, e che vi mando registrato su cassetta, a volte funziona e a volte no. Potete spiegarmene il motivo?
(Fulvio F. - Genova)
(G. Zinnati - Partanna)

• Come avrai forse già intuito, leggendo una risposta pubblicata sul numero 52, il motivo è da ricercarsi nell'approssimazione intrinseca del calcolatore (C/64) che possiedi.

Il programma del nostro lettore, infatti, genera casualmente alcuni numeri Random, li elabora supponendo che si riferiscano a determinate unità di misura e ne chiede la conversione in altre unità (da litri a decalitri, da centimetri a pollici e così via).

In questo lavoro, soprattutto elaborando valori decimali, può capitare l'esigenza di un'approssimazione particolare che induce il computer in errore. L'utente, che ritiene di aver risposto in maniera corretta alla domanda posta, non può che constatare l'imprecisione.

Però, e mi rivolgo a Fulvio, non ti pare che, a parte tutto, il programma sia un po' troppo lungo (quasi 32 K!) per le funzioni che svolge? Ti consiglio di studiare metodi di ottimizzazione, in modo da risparmiare tempo e, soprattutto, pazienza.

PROTESTANTI

□ Aggiungo la mia alle proteste di altri lettori che, a ragione, si lamentano per lo scarso spazio destinato al C/16, e mi permetto di inviarvi due programmi per questo "sfortunato" computer che, a mio parere, sono molto validi.
(Giovanni P. - None)

• Una tirata d'orecchie al nostro bravo ma distratto amico: questa punizione non viene inflitta per il suo desiderio di soccorso nei confronti del C/16, ma solo perchè ha inviato due programmi ("Old" e "Espansione di memoria") che sono già stati pubblicati più di una volta (pur se in forma diversa da quella inviata) sul nostro periodico. Se davvero ami il tuo computer dovresti leggere con più attenzione le riviste che acquisti, se non altro per risparmiar fatica nell'inventare programmi già divulgati.

Ne approfitto, comunque, per sottolineare che questa è l'unica lettera di protesta che sia giunta in redazione da un po' di tempo a questa parte. La mancanza di lamentele è da attribuire, a mio modesto parere, al notevole spazio che, di recente, è stato destinato al C/16 ed al C/128.

La nuova rubrica che ha preso il posto dell'enciclopedia di Routine è, in effetti, una vera e propria sezione destinata agli utenti di C/16 e C/128; credetemi: di più non si può fare, e me ne dispiace perchè io stesso preferisco il C/128 al C/64!

VECCHIAIA

□ Posseggo un sistema completo Commodore Pet (4032, 4040, 4022) acquistato nel 1981. Poichè non lo utilizzo più per scopi professionali, mi piacerebbe contattare un Club, anche se straniero, con cui scambiar giochi e programmi vari. Pensate che sia possibile?
(Edoardo Cardone - S.Severo)

• Purtroppo non penso che al mondo vi sia ancora qualcuno che utilizzi il validissimo ma ormai obsoleto computer.

Se il calcolatore è corredato anche da vari programmi di contabilità, o altri package professionali per cui il Pet era utilizzato, l'unico consiglio è quello di vendere l'intero sistema Hard/Soft a qualche conoscente, in modo da acquistare, con il ricavato, un C/64 ed un drive; dubito, infatti, che un sistema Pet, pur se completo, possa esser venduto ad un prezzo troppo distante dal milione di lire.

GRAZIE

□ Tutto ciò che ho imparato sul l.m. lo devo a voi e mi piacerebbe veder pubblicato questo mini-programma, per C/64, che permette di modificare il messaggio "Ready." a piacimento (purchè la parolina sostitutiva sia lunga non più di sei caratteri).

P.S.: inoltre vorrei sapere, per curiosità, se...
(Sandro Rizzetto - Bolzano)

• Il simpatico programma non fa altro che trascrivere, nella Ram posta sotto la Rom del Basic, l'intero linguaggio interprete. In seguito (riga 20) seleziona il banco Ram "oscurando" la Rom. Inutile dire che, a partire da 41848, è presente la parolina "Ready."

```
10 for i=40960 to 49151: poke i,
peek(i):next
20 poke 1,54
30 x=41848
40 for k=x to x+5:read a:poke
k,a:next
50 data 67,46,67,46,67,46
```

P.S.: no, l'Alfasud non l'ho ancora cambiata; però sto pensando di prendere un'Alfa 75 turbodiesel. Che ne dici, faccio bene?

IL MIGLIORE

□ Possiedo un C/64 con drive 1541 e stampante 801; vorrei sapere se esiste un w/p migliore di quello che possiedo e che uso di solito.
(Andrea Renzi - Roma)

• Purtroppo non dici qual'è il w/p in tuo possesso, ma ritengo che il motivo della insoddisfazione sia dovuto alla stampante che, senza alcun dubbio, è valida per lavori hobbystici, ma non è certo in grado di svolgere lavori di un certo pregio (lettere commerciali, tesi, fatture e così via).

Se, però, non sei contento di Easy Script, che personalmente ritengo il migliore w/p per C/64, dubito che altri w/p possano dare maggiori soddisfazioni; per "salire" di livello, a mio parere, è necessario cambiar sistema: un Ms-Dos compatibile, oppure (perchè no?) l'Amiga.

KOALA SU STAMPANTE

□ Come è possibile far stampare una schermata di Koala su una stampante Mps-803?
(Orlando Proietti - Roma)

• Il notissimo programma Koala consente di sfruttare le capacità grafiche del C/64 per effettuare diversi tipi di disegni, naturalmente a colori.

La tecnica usata, però, è quella multicolor che, come è noto, consente di gestire quattro colori (invece di due) in ciascuna cella di 8x8 pixel. La pagina grafica, in realtà, è il risultato di una complessa elaborazione che vede in azione 8000 byte della "solita" pagina grafica, 1000 byte per la pagina dei primi due colori ed altri 1000 per elaborare gli altri due.

Purtroppo tale modo di operare ha un preciso significato solo per il computer; ciò che, invece, viene di solito trasferito con i normali programmi di hard copy è la sola pagina grafica la quale, però, non rappresenta più un'immagine intelleggibile.

Per traferire su stampante, quindi, un disegno realizzato con il Koala, è necessario scrivere un programma che riesca ad individuare, cella per cella, l'esatto significato dei colori e ne "estragga" la gradazione di grigio ideale.

Esistono numerose routine (naturalmente in l.m.) alcune delle quali contenute in cartucce di vario tipo.

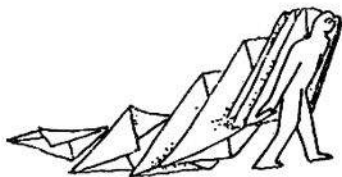
Non è detto che, prima o poi, anche noi ci decideremo a divulgare una procedura che svolga la funzione richiesta.

FUTURO DEL GEOS

□ Pensate di trattare, in futuro, argomenti inerenti il sistema operativò Geos?
(Illeggibile - Genova)
(Angelo Cordaro - Sciacca)

• Il Geos, appunto, è un sistema operativo, contenuto su disco, che si sostituisce al s.o. originale del C/64 e ne potenzia le funzioni (maggior velocità, soprattutto) facendo anche ricorso al mouse.

Non penso che ci sia poi tanto da dire, anche perchè viene fornito gratis ad ogni acquirente di C/64 corredato di istruzioni, per di più, in italiano.

Semmai può essere interessante pubblicare, di tanto in tanto, recensioni di package che girano, come si dice in gergo, "sotto" Geos. Sembra infatti che, finalmente, qualcosa si stia realizzando: staremo a vedere.

COMPI-DECOMPI-LARE

□ Per aumentare la velocità dei programmi che scrivo in Simon's Basic vorrei compilarli; esiste un compilatore adatto? Tempo fa compilai alcuni programmi con il Pet-Speed ma, purtroppo, ho inavvertitamente cancellato i listati sorgenti: esiste un de-compilatore per recuperarli?
(Francesco Zambuto - Agrigento)

• Ahimè, confesso la mia... disinformazione; non mi pare, tuttavia, di aver mai sentito parlare di compilatori per Simon's Basic. Mi capitò di avere notizie di un decompilatore per Pet Speed che, però, non ho mai avuto la necessità di usare (e, di conseguenza, di tentare di procurarmelo).

C'è qualche lettore che possiede le utility richieste? Se dovesse mandarle, infatti, le inserirei certamente in "Directory", a patto, ovviamente, che siano di dominio pubblico.

Aiutateci a servirvi meglio

• Spesso alcuni lettori, che dichiarano di possedere numeri arretrati della nostra rivista, porgono quesiti le cui risposte sono già state esplicitamente pubblicate (in occasione di risposte ad analoghe domande) oppure sono contenute in articoli presenti nei fascicoli in loro possesso.

Per evitare di ripetere argomenti già trattati, pertanto, ricordate di indicare sempre, nelle lettere che ci inviate, i numeri dei fascicoli in vostro possesso: potremmo infatti indicarvi, se esistono, gli articoli che, in un modo o in un altro, possono chiarire gli argomenti richiesti.

AMIGA

UN'ICONA PER TUTTI

□ Ad alcuni programmi vorrei assegnare la specifica icona, di cui attualmente sono privi, in modo da caricarli clickandoli con il mouse.
(Maurizio Mirabelli - Tessera)

• Il programma di utilità del Workbench "IconED" serve anche ad assegnare icone ai programmi che non ne dispongono.

Basta creare l'icona desiderata e salvarla, specificando il nome completo...

df0:progs/pippo

...con il nome del programma a cui deve essere assegnata. Iconed aggancia automaticamente il suffisso ".info". Ma attenzione: non tutti i programmi possono essere eseguiti da Workbench clickando col mouse su un'icona realizzata nel modo descritto perchè il programma deve essere "predisposto" per essere usato da Workbench.

Ad esempio, i comandi della directory C di Workbench, anche se è assegnata un' icona, non vengono caricati. Segnaliamo comunque ai lettori, amanti degli ideogrammi di Intuition, la disponibilità del nostro dischetto "1000 icone", che incorpora anche una nuova utility di creazione e qualche nuova informazione.

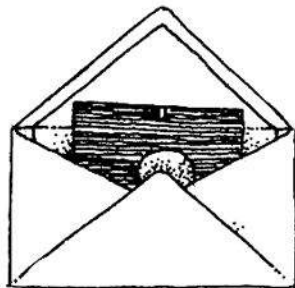
UN BELL'OROLOGIO

□ E' possibile fare in modo che l'orologio (C.C.C. n. 49) possa comparire come finestra nel mio C/16 durante l'esecuzione di altri programmi? Come è possibile far apparire numeri progressivi sul C/16?
(M.Gabriella Mai - Avezzano)

• Purtroppo la richiesta, per essere esaudita, richiede la disponibilità di una gran quantità di memoria e, se non bastasse, anche di una elevata velocità di elaborazione dal momento che dovrebbe, inevitabilmente, lavorare nel ciclo di Interrupt. Il programma cui fai cenno serviva solo a scopo didattico, per far comprendere l'uso di alcune istruzioni grafiche specifiche del Basic 3.5 (del C/16, appunto).

La seconda domanda, invece, è piuttosto semplice:

```
100 Input "Parto da";P
110 Input "Intervallo";I
120 Print P: P=P+I
130 Goto 120
```

Naturalmente questa è solo l'idea base: tocca a te elaborare il concetto per trasportare il procedimento nei tuoi listati.

ERRORE

□ Vorrei segnalare quello che, a mio avviso è un errore presente nel listato di pag. 44 (C.C.C. n. 51).
(Giulio Pompei - Milano)

• Il nostro lettore ha perfettamente ragione: la parte di riga 790 in cui compare...

"Media: "W

...deve esser così modificata:

"Media: " W/(I-1)

Ne approfitto per scusarmi dell'errore involontario che, mi auguro, sia stato prontamente individuato anche dagli altri lettori interessati al programma in oggetto.

PROTEZIONE FATALE

□ Pensavo che, via software, fosse impossibile rovinare i circuiti integrati del C/64. In un vostro recente articolo, invece, affermate che è possibile bruciare un chip attivando un loop infinito. Devo preoccuparmi?
(Dino Calvisi - Petogna)

• Un circuito integrato è un circuito elettronico come tutti gli altri e, come tale, soggetto a ben precisi limiti di funzionamento.

In teoria i c.i. dovrebbero esser progettati in modo tale da sopportare le correnti che li attraversano, senza presentare inconvenienti di alcun genere.

Può capitare, tuttavia, che alcune giunzioni elettroniche, a livello microcircuitale, siano più deboli delle altre; ne consegue che se la corrente che vi scorre ha una durata superiore al previsto è possibile una cattiva dispersione del calore che potrebbe (il condizionale è d'obbligo, soprattutto nei chip progettati in modo poco accurato) tradursi in una rottura locale del chip stesso.

Naturalmente quella raccolta era una voce di corridoio, poco più di un pettegolezzo, che è stato riportato sulla nostra rivista con il solo beneficio d'inventario. Ti posso assicurare, tuttavia, che questo fantomatico chip non appartiene al C/64.

LASERNET PER C/16?

☐ Se, come affermate, presso la Lasernet è disponibile software per C/16, dovrebbe corrispondere la vendita di un modem (o adattatore telematico) per C/16, altrimenti che senso avrebbe proporre s/w per C/16 agli utenti di C/64?
(Arrigo Frigeni - Bergamo)
(Roberto Nerozzi - Bologna)

• E' vero, non ci avevamo pensato. Purtroppo, però, la commercializzazione di un apparecchio telematico specifico per C/16 è al di fuori della realtà commerciale e, di conseguenza, improponibile.

Mi rendo conto che questa risposta possa apparire sconfortante, ma non dimentichiamo, tuttavia, che per ammortizzare le spese di studio, realizzazione e commercializzazione di un simile prodotto sarebbe necessaria una ragionevole certezza di vendita di parecchie migliaia di apparecchi.

Il che non è.

CONCORSI

L'A.I.C.A. (Associazione Italiana per l'Informatica ed il Calcolo Automatico), e lo SMAU di Milano (Salone Internazionale per l'Ufficio e l'Automazione) bandiscono il concorso "Computer Play" che quest'anno ha per tema "Il calcolatore come strumento amichevole".

I lavori, di software o di hardware, sviluppati da giovani di età inferiore ai 21 anni, verranno esaminati e premiati nel corso della prestigiosa manifestazione fieristica milanese (29 settembre - 3 ottobre) con lo scopo di realizzare un punto di contatto tra l'industria ed i giovani, soprattutto studenti.

Il termine di presentazione dei lavori è il 31 agosto; gli argomenti dovranno vertere su:

- Software: simulazione, grafica, suono, sistemi esperti, comunicazione.
- Hardware: bracci meccanici, robot, attuatori.

Per ulteriori informazioni:

Studio Giornalistico EDIT
Via Antonio da Recanate, 4
20141 Milano
Tel: 02/ 66.93.161
02/ 66.92.681

EXTRA IGNORED

☐ A differenza degli altri errori, quello che emette il messaggio "Extra ignored" non viene intercettato dal comando Trap del mio C/128. E' possibile ovviare all'inconveniente?
(Antonio B.)

• Il messaggio "Extra ignored" viene emesso, in fase di Input, quando il numero di dati digitato è maggiore di quello richiesto. Ad esempio, con il seguente programma...

```
100 Input "Scrivi un dato"; A$
110 Print A$: Goto 100
```

... rispondendo, alla domanda di riga 100, con un valore numerico o con un gruppo di caratteri alfanumerci, ciò che viene digitato viene riscritto grazie al Print di riga 110.

Se, però, si digitano gruppi di caratteri all'interno dei quali è presente un carattere di doppio punto (:) oppure di virgola (,) come ....

82764,876

...oppure...

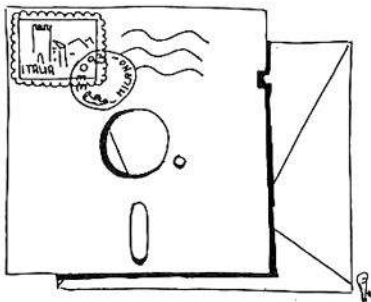
hf : jouhgf

...verranno associati, alla variabile A$, solo i caratteri che precedono i segni di punteggiatura battuti. Gli altri (cioè gli "Extra") verranno ignorati perchè il Basic considera il segno di punteggiatura come fine di invio dati.

Contemporaneamente viene emesso un messaggio che ha lo scopo di avvertire l'utente che non è stato possibile elaborare i dati "successivi".

Il messaggio, pertanto, non è considerato un vero e proprio errore, dal momento che rappresenta, appunto, un semplice avvertimento e non viene, di conseguenza, intercettato da Trap. Tuttavia la presenza di un simile messaggio può "sporcare" un'eventuale maschera video, faticosamente costruita dal programmatore.

Non rimane che ricorrere ad una delle tantissime routine di Input controllato che, in realtà, sono routine affidate all'istruzione Get (e non Input) che, carattere dopo carattere, riescono a capire se il tasto premuto deve esser accettato oppure no.

Un altro sistema può essere quello di scandire l'area buffer (512 - 600)

del C/64 relativa alla zona di memoria in cui vengono accumulati i tasti battuti prima di essere elaborati dalla routine di Input. In quest'area, infatti, vengono trascritti i valori dei tasti secondo un codice particolare: a 0 corrisponde il tasto Return, a 44 la virgola e a 58 il doppio punto.

Il semplice listato, da studiare e sviluppare a parte, può chiarire meglio le idee:

```
100 input "scrivi un dato";a$: x=0
110 for i=512 to 600: y=peek(i)
115 if y=0 then print tab(20) "senza extra": i=600: goto 130
120 if y=44 or y=58 then print tab(20) "con extra": i=600
130 next
140 print a$: goto 100
```

Sarà sufficiente esaminare il contenuto di tale area Ram e stabilire, nel caso in cui prima di uno zero sia presente 44 oppure 58, se attivare, o meno, una routine che provveda a cancellare il messaggio "Extra Ignored" apparso.

AMIGA

PUNTUALIZZIAMO

□ Come può, l'Amiga, distinguere un disco copia dall'originale? (ed altre domande)
(Giuseppe Caruso - Arezzo)

• Il sistema operativo (S.O.) di Amiga è solitamente in grado di distinguere due dischi, pur se dotati dello stesso nome, grazie alla data di formattazione impressa, indipendentemente dal fatto che si formatti, o meno, con l'opzione NOICONS che provvede solo ad inserire file supplementari (disk.info, trashcan e trashcan.info).

Diciamo "solitamente" perchè qualche volta capita, per difetti di AmigaDOS od uso di programmi con imperfetta gestione di AmigaDOS, che anche il sistema si confonda. Comunque si tratta di condizioni limite, visto che solitamente ogni dischetto ha un nome diverso.

• Per ciò che riguarda l'AmigaBASIC è necessario tener presente che, a differenza del Basic del C/64, richiede sempre almeno uno spazio tra le parole-codice. Non si può scrivere...

```
fori=1to1000:printa:nexti
```

...ma...

```
for I =1 to 1000: print A: next I
```

Ricordiamo anche che AmigaBASIC, in fase di edizione, trasforma automaticamente in maiuscolo le parole riservate (PRINT, DIM, REM) e lascia in minuscolo le altre.

• Passando ad altro, uno dei problemi più comuni riscontrati nella gestione dei file è quello relativo all'impossibilità di salvare file di programma e di Notepad, perchè nel nome del file si deve sempre specificare il disco in cui si vuole scrivere (o da cui caricare): suggeriamo di rileggere gli articoli che riguardano AmigaDOS (pubblicati a partire dal numero 42).

• Infine: non è possibile "visualizzare il listato" di programmi come i Demos od IconED perchè sono scritti in linguaggio C compilato (cioè in linguaggio macchina), a meno di avere una stampante per scrivere le migliaia di linee di linguaggio macchina di cui sono composti ed un programma disassemblatore.

Se non bastasse, si capisce ben poco del listato stampato se non si dispone del testo sorgente originale.

QUANTA MEMORIA?

□ Che quantità di memoria è contenuta in un blocco di disco?
(Turk 182)

• Il dischetto di un drive 1541 è capace di memorizzare fino a 664 blocchi di 256 byte ciascuno per un totale di 169984 byte.

In pratica, però, la quantità di memoria racchiusa in un floppy è quasi sempre minore di tale ragguardevole cifra.

Quando, infatti, si registra su disco un programma, oppure un file sequenziale, l'allocazione avviene per blocchi di 256 byte alla volta.

Se, ad esempio, un programma super-breve è costituito da soli 100 byte occuperà la stessa quantità di blocchi (cioè 1) che è richiesta per memorizzare un programma di 30 byte oppure di 256 byte oppure, nel caso di un file sequenziale, di un solo byte!

Ciò avviene perchè i dati sono organizzati per "pagine" di 256 byte che non possono essere occupate parzialmente: nel caso in cui un programma occupi solo alcuni byte all'interno di un blocco, gli altri byte rimasti inutilizzati non possono esser recuperati, se non a prezzo di complicate gestioni particolari.

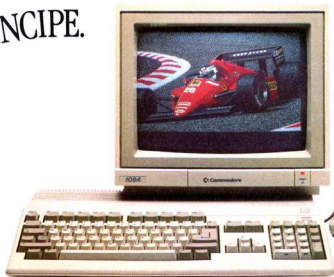
Ecco spiegato perchè, ad esempio, due file sequenziali, notevolmente diversi per ciò che riguarda il contenuto (il primo di 257 byte ed il secondo di 512, cioè quasi il doppio), occupano, in entrambi i casi, due blocchi di disco.

Amiga 500

è il principe dei computers, fiore all'occhiello della grande dinastia Commodore.
Oggi puoi portarti a casa questo gioiello dell'informatica a condizioni estremamente vantaggiose: perché Commodore, se scegli un'Amiga 500, valuta il tuo vecchio computer ben **200.000** lire se è un C 64 e 100.000 lire se è un altro modello Commodore o un'altra marca.
Queste valutazioni saranno ridotte alla metà se il tuo vecchio computer non è funzionante o è incompleto.
Amiga 500 ti viene proposto in una scatola kit completa di modulatore e scrigno del software.
Corri col tuo vecchio computer nel più vicino Commodore Point o in un negozio autorizzato all'operazione "Cambia con il Principe": Amiga 500 è lì che ti aspetta.

CAMBIA IN
Commodore

200.000 LIRE.
PER IL TUO VECCHIO COMPUTER

TE LO CAMBIAMO CON IL PRINCIPE.

AMIGA 500. COMPUTER DELL'ANNO 1987.

ECCO I COMMODORE POINT CHE CAMBIANO IL TUO VECCHIO COMPUTER CON IL PRINCIPE:

Commodore

Lombardia

AL RISPARMIO di ADA CASTOLDI e SALA S.
V.le Monza, 204 - MILANO

BRAHA ALBERTO s.d.f.
Via Pier Capponi, 5 - MILANO

E.D.S. ELECTRONIC DATA SYSTEMS s.r.l.
C.so Porta Ticinese, 4 - MILANO

E.S.C. s.r.l.
Via Roggia Scagna, 7 - MILANO

FAREF S.p.A.
Via A., Volta, 21 - MILANO

GBC ITALIANA S.p.A.
Via Petrella, 6 - MILANO
Via Cantoni, 7 - MILANO
V.le Matteotti, 66 - CINISELLO BALSAMO

GIGLIONI LAURA
Via D'Ovidio, 8 - MILANO

GIGLIONI s.r.l.
V.le L. Sturzo, 45 - MILANO

LOGITEK s.r.l.
Via Golgi, 60 - MILANO

MARCUCCI S.p.A.
Via F.lli Bronzetti, 37 - MILANO

MELCHIONI S.p.A.
Via P. Colletta, 37 - MILANO

MESSAGGERIE MUSICALI S.p.A.
Galleria del Corso - MILANO

NEWEL s.r.l.
Via Mac Mahon, 75 - MILANO

RIVOLA s.n.c.
Via Vitruvio, 43 - MILANO

F.LLI GALIMBERTI s.a.s.
Via Nazionale dei Giovi, 28/35 BARLASSINA (MI)

CASA DELLA MUSICA s.a.s.
Via Indipendenza, 21 - COLOGNO MONZESE (MI)

PENATI s.r.l.
Via Verdi, 28/30 - CORBETTA (MI)

EPM System
V.le Italia, 12 - CORSICO (MI)

P. GIORGIO OSTELLARI
Via Milano, 300 - DESIO (MI)

CENTRO COMPUTER PANDOLFI
Via Corridoni, 18 LEGNANO (MI)

COMPUTER s.a.s. di DE ANDREIS & C.
Via Vecellio, 41 - LISSONE (MI)

FUTURA s.a.s.
Via Solferino, 31 - LODI (MI)

MBM - INFORMATICA SYSTEMS S.p.A.
C.so Roma, 112 - LODI (MI)

L'AMICO DEL COMPUTER s.a.s.
V.le Lombardia, 17 MELEGNANO (MI)

BIT 84 s.a.s.
Via Italia, 4 - MONZA (MI)

I.C.O. di Giorgio Odorici
Via dei Tigli, 14 - OPERA (MI)

COMIF
Via Autolinee, 10 - BERGAMO

CORDANI S.p.A.
Via dei Caniana, 8 - BERGAMO

D.R.B. di De Ruschi Ernesto
Via Borgo Palazzo, 6 - BERGAMO

NEW SYSTEMS di Mazza Carlo R. & C.
Via Paglia, 36 - BERGAMO

COMPUTER TEAM hi-tec s.r.l.
Via Verdi, 1/B - CARVICO (BG)

OTTICO OPTOMETRISTA ROVETTA di Vasco Vasconi
P.zza Garibaldi, 6 - LOVERE (BG)

A.I.S. International s.r.l.
Via San Carlo, 25 - SAN PELLEGRINO TERME (BG)

SISTHEMA s.r.l.
Via Roma, 45 - SARNICO (BG)

COMPUTER CENTER
Via Cipro, 62 - BRESCIA

INFORMATICA 2000
Via Stazione, 16/B - BRESCIA

VIGASIO MARIO
Portici Zanardelli, 3 - BRESCIA

MISTER BIT
Via Mazzini, 70 - BRENO (BS)

CAVALLI PIETRO
Via 10 Giornate, 14B CASTREZZATO (BS)

VIETTI GIUSEPPE
Via Milano, 1/B - CHIARI (BS)

MEGABYTE di Trabucchi
P.zza Duomo, 17 - DESENZANO DEL GARDA (BS)

DITTA BARESI RINO & C. s.n.c.
Via XX Settembre, 7 - GHEDI (BS)

INFO CAM s.r.l.
Provinciale 38 GRATACASOLO (BS)

KOFBAKER s.r.l.
Via Marchi, 65/B VESCOVATO (BS)

2M ELETTRONICA s.r.l.
Via Sacco, 3 - COMO

IL COMPUTER di Ferrari Catia
Via Indipendenza, 90 - COMO

ELTRONGROS S.p.A.
Via L. Da Vinci, 54 BARZANÒ (CO)

EGA di Giuseppe Abello
Via Mazzini, 42 CASSAGO BRIANZA (CO)
Filiale: Via Aldo Moro, 17 GALBIATE (CO)

DATA FOUND computer shop
Via A. Volta, 4 - ERBA (CO)

CIMA ELETTRONICA s.a.s.
Via Leonardo da Vinci, 7 - LECCO (CO)

FUMAGALLI
Via Cairoli, 48 - LECCO (CO)

RIGHI ELETTRONICA
Via G. Leopardi, 26 OLGIATE COMASCO (CO)

MONDO COMPUTER
Via Giuseppina, 11/B - CREMONA

TELCO di Gianfranco Zambiasi & C. s.n.c.
Piazza Marconi, 2/A - CREMONA

PRISMA s.n.c.
Via Buoso da Dovara, 8 CREMONA

ELCOM/GBC
Via IV Novembre, 56/58 CREMA (CR)

EUROELETTRONICA
Via XX Settembre CREMA (CR)

COMPUTER s.a.s. di Tubaldo Elso & C.
Galleria Fermi, 7 - MANTOVA

ELETTRONICA DI BASSO
Viale Risorgimento, 69 MANTOVA

32 BIT (Computer Studio)
Via Cesare Battisti, 14 MANTOVA

POLIWARE s.r.l.
Corso Carlo Alberto, 66 - PAVIA

LOGICA INFORMATICA s.r.l. computer shop
V.le Monte Grappa, 32 VIGEVANO (PV)

M. VISENTIN
C.so Vittorio Emanuele, 76 VIGEVANO (PV)

COMPUTER LINE
Via G. Carducci, 4 - PIACENZA

DELTA COMPUTER
Via Martiri della Resistenza, 15/4 PIACENZA

SOVER s.n.c.
Via IV Novembre, 60 PIACENZA

CIPOLLA MAURO
Via Tremoggia, 25 - SONDRIO

FOTONOVA di Bianchi Rita & C. s.n.c.
SAN PIETRO DI BERBENNO (SO)

DIMECO SISTEMI s.n.c.
Viale Garibaldi ang. Iva VARESE

IL CENTRO ELETTRONICO s.r.l.
Via Morazzone, 2 - VARESE

SUPERGAMES s.a.s. di Ranzoni Franco
Via Carrobbio, 13 - VARESE

BUSTO BIT di Vittorio Ornago
Via Gavinana, 17 - BUSTO ARSIZIO (VA)

CRESPI GIUSEPPE & C. s.n.c.
V.le Lombardia, 59 CASTELLANZA (VA)

COMPUTER SHOP s.n.c.
Via A. Da Brescia, 2 GALLARATE (VA)

LIMA IMPORT EXPORT s.r.l.
c/o Grandi Magazzini Bossi Via Clerici, 196 GERENZANO (VA)

I.A.C. nuove tecnologie s.n.c.
Via Matteotti, 38 SESTO CALENDE (VA)

Piemonte

ABA ELETTRONICA s.n.c.
Via C. Fossati, 5/P - TORINO

ALEX COMPUTER E GIOCHI di Bovolenta Diana
C.so Francia, 333/4 - TORINO

COMPUTER HOME
Via San Donato, 46/B - TORINO

COMPUTING NEW s.n.c. di Chiricosta Paolo & C.
Via Marco Polo, 40/E - TORINO

DE BUG di Gughelmotto Roberto
C.so Vittorio Emanuele II, 22 TORINO

DESME UNIVERSAL s.a.s.
Via San Secondo, 95 - TORINO

F.D.S. ALTERIO di Fernando De Siati
Via Borgaro, 86/D - TORINO

INFORMATICA ITALIA s.r.l.
C.so re Umberto, 129 - TORINO

MT INFORMATICA s.r.l.
C.so Giulio Cesare, 58 - TORINO

NEW BUSINESS COMPUTER s.a.s.
Via Nizza, 45 - TORINO

RADIO TV MIRAFIORI di Maggiani e Barile s.n.c.
C.so Unione Sovietica, 381 TORINO

SMT ELETTRONICA s.n.c. di S. Patrucco & C.
Via Bibiana, 83/B - TORINO

PAUL E CHICO VIDEOSOUND di Varetto F. & C. s.n.c.
Via Vittorio Emanuele, 52 CHIERI (TO)

BIT INFORMATICA
Via V. Emanuele, 154 CIRIÈ (TO)

HI-FI CLUB
C.so Francia, 92/C COLLEGNO (TO)

I.C.S. s.r.l.
Stradale Torino, 73 IVREA (TO)

CERUTTI s.n.c.
Corso Torino, 234 PINEROLO (TO)

EUREX di Ponchia Rag. Gianfranco & C. s.a.s.
C.so Indipendenza, 5 RIVAROLO C.SE (TO)

BIT MICRO di Gras Ginette
Via Mazzini, 102 - ALESSANDRIA

WEST RECORDS (soc. Servizi Informatici S.a.s.)
C.so Roma, 85 - ALESSANDRIA

S.G.E. ELETTRONICA di Solarolo Giuseppe
Via Bandello, 19 - TORTONA (AL)

RECORD di Fiorina Giuseppe
C.so Alfieri, 166/3 (Galleria Argenta) - ASTI

ROSSI COMPUTERS di Rossi Metello & C.
C.so Nizza, 42 - CUNEO

PUNTO BIT s.n.c. di Vacca Giovanni & C.
C.so Langhe, 26/C - ALBA (CN)

SDI s.n.c.
Via Vittorio Emanuele, 250 BRA (CN)

CURETTI AUGUSTO
C.so Italia, 3 - Mondovì (CN)

ELCOM s.r.l.
C.so Mazzini, 11 - NOVARA

PROGRAMMA 3 s.n.c.
V.le Buonarroti, 8 - NOVARA

PUNTO VIDEO di La Vecchia Domenico
C.so Risorgimento, 39 NOVARA

COMPUTER di Mirco Polacco & C. s.a.s.
Via Monte Zeda, 4 - ARONA (NO)

ALL COMPUTER di Peirotti Alberto
C.so Garibaldi, 106 BORGOMANERO (NO)

MICROLOGIC s.a.s.
Via Giovanni XXIII, 2 DOMODOSSOLA (NO)

ELLIOTT COMPUTER SHOP s.r.l.
Via Don Minzoni, 32 INTRA (NO)

DITTA ELETTROGAMMA s.a.s.
C.so Bormida ang. Via Montanara VERCELLI

ELETTRONICA s.a.s. di Bellamo A. & C.
Strada Torino, 15 - VERCELLI

C.S.I. TEOREMA s.r.l.
Via Losana, 9 - BIELLA (VC)

FOTOSTUDIO TREVISAN
Via XXV Aprile, 24/B CORSATO (VC)

STUDIO FOTOGRAFICO IMARISIO di Imarisio Roberto
P.zza Martiri Libertà, 7 TRINO (VC)

AMIGA

NON COPIA

□ Con il mio Amiga non riesco ad effettuare copie servendomi di un drive compatibile esterno, mentre riesco benissimo a svolgere il compito usando lo stesso drive, ma collegato all'Amiga di un mio conoscente.
(Umberto Zanatta - Preganzion)

• I problemi di copiatura di programmi (birichini!) derivano dal fatto che le meccaniche dei drive interni di Amiga sono di qualità modesta, spesso inferiori a quelle dei drive supplementari, anche se costruiti da ditte italiane o da "cantinari".

Inoltre potrebbero esserci piccoli problemi di allineamento della testina (del drive interno del tuo Amiga), nel qual caso o rinunci a copiare i programmi "critici", oppure provvedi a far allineare il drive presso un centro specializzato.

COMPILARE IN DUE TEMPI

□ Un mio programma Basic è, in effetti, costituito da due parti: la prima è sempre fissa, mentre la seconda, per come è fatta, deve spesso essere aggiornata.
E' possibile compilare la prima parte e, in seguito, compilare solo gli aggiornamenti da "appendere" alla parte fissa?
(G. Luigi Curti - Mortara)

• Purtroppo, al momento della compilazione di un programma Basic, la procedura organizza la memoria Ram disponibile in un blocco unico nel quale è ben definita la zona riservata alle variabili, la zona relativa all'esecuzione e così via.

Non è assolutamente prevista un'"uscita" che, invece, potrebbe favorire la procedura accennata.

Si può, tuttavia, provare a dividere il listato in due programmi, del tutto separati l'uno dall'altro. Quindi inserire, nel primo, il comando Load relativo al secondo programma. In seguito compilare separatamente i due programmi assegnando, al secondo, il nome indicato dal Load del primo. Si consiglia un attento studio del caricamento in Overlay descritto nei due inserti ("Viaggio nel Basic") pubblicati sui numeri 43 e 44 di C.C.C.

In questo modo si potrebbe tentare di realizzare la procedura richiesta, purchè non siano presenti istruzioni "strane" che possano entrare in conflitto con le routine del compilatore.

Un altro modo di risolvere la faccenda potrebbe essere quello di gestire i vari aggiornamenti con file sequenziali, da leggere sempre allo stesso modo dall'unico programma principale.

SENSAZIONALISMI

□ Leggendo alcune riviste di settore mi sono confuso le idee e chiedo delucidazioni su varie notizie relative all'Amiga (segue elenco).
(Egidio Amedei - Salerno)

• Le redazioni di alcune pubblicazioni, pur di attirare i lettori, decidono di riempire la copertina con notizie ritenute interessanti (del tipo: "Aggredita sotto la pioggia, viene costretta a reggere l'ombrello durante l'aggressione").

Molto spesso, però, non si accorgono di gettare scompiglio nelle tremebonde menti dei potenziali acquirenti, con la conseguenza, certamente indesiderata, di creare diffidenza verso i prodotti commercializzati.

Per ciò che riguarda presunte nuove versioni del sistema operativo di Amiga, sembra (ripeto: sembra) che esista una versione sufficientemente aggiornata che, però, può esser caricata solo sui vecchi modelli Amiga 1000, dal momento che sia il Kickstart che il Workbench vanno caricati da disco; i modelli 500 e 2000, invece, hanno il Kickstart residente su Rom e la versione di cui parlo non può esser caricata.

Non va dimenticato, ad ogni buon conto, che tutti i S.O. di questo mondo sono stati aggiornati, in modo spesso sostanziale, con il passare del tempo (vedi CP/M ed Ms/Dos). Il vantaggio di questi computer, anzi, consiste proprio nel non costringere la gente a cambiar computer nel caso di successive versioni di sistemi operativi: è sufficiente cambiar dischetto.

Nel caso, finora mai verificato, in cui una nuova versione di S.O. risultasse incompatibile con il software già disponibile, la faccenda si risolverebbe in un modo molto semplice: con il s/w vecchio si caricherebbe il S.O. vecchio, con il s/w nuovo, quello più recente.

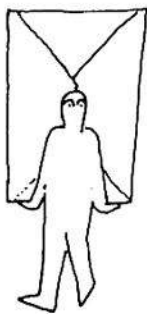
Sarebbe seccante, invece, esser costretti a sostituire la Rom del Kickstart nel caso si verificassero incompatibilità tra vecchie e nuove versioni del Workbench.

Con questo voglio solo dire che non è necessario attendere la versione "definitiva" del S.O. di Amiga: nell'informatica non c'è nulla di definitivo (IBM insegna). Quando verranno realizzate nuove versioni di S.O. basterà andare dal più vicino

computeraio per procurarsele (magari gratis).

Per ciò che riguarda i nuovi monitor, puoi dare uno sguardo ai listini pubblicati in fondo alla rivista. Una tale varietà di modelli, lungi dal voler confondere le idee, indica, al contrario, la volontà della Commodore di accontentare le diversificate esigenze degli utenti.

Per esaminare da vicino le altre potenzialità di Amiga (sviluppabili con le numerose schede che, un po' per volta, verranno proposte) basterà avere la pazienza di attendere il prossimo SMAU di settembre.

Oppure di seguire la nostra pubblicazione: si tenga presente, e lo ripetiamo ancora una volta, che su queste pagine vedrete sempre e solo notizie relative a schede ed accessori reperibili (quasi) ovunque in Italia. I senzazionalismi frustranti non ci interessano.

NON FUNGE

□ Seguo alla lettera le istruzioni riportate nelle note relative al funzionamento dell'enciclopedia di routine l.m. ma, purtroppo, non mi riesce di registrare in un unico file le varie routine (ed altre domande).

(Giancarlo Bonfieni - Brescia)

• Quasi certamente l'errore è da addebitare alla parziale sovrapposizione delle nuove routine con quelle vecchie (che cessano, in tutto o in parte, di funzionare); oppure all'errata trascrizione del programma "Save zona Ram" che altera in modo inadeguato i puntatori di inizio e fine dell'area Ram da registrare (mancando di salvare una parte delle routine faticosamente trascritte).
• Il termine del comando While (Gw-Basic) è individuato dal carattere di doppio punto (:). In sua assenza le istruzioni successive non vengono ri-

AMIGA

MOLTE DOMANDE

□ Vorrei alcune informazioni su determinati argomenti (segue elenco).

(Sergio Benedetti - Torre del Lago)

• Il Transformer per Amiga funziona, teoricamente, con tutti i programmi che non fanno uso di hardware aggiuntivo (schede grafiche, eccetera). In realta, come hai notato anche tu, molti programmi non girano, per cui bisogna rivolgersi all'Amiga 2000 con scheda emulatrice (ed allora si ha una vera compatibilità al 100%); oppure si usa il Transformer ultima versione (1.2) solo per scopi didattici.
• Abbiamo avuto modo di esaminare la versione 1.3 del DOS di Amiga, giunto alla versione Beta 0.65 (ma non definitiva), per cui sarà probabilmente disponibile agli inizi di questa estate (sperando di vederla, comunque, allo SMAU di settembre).

Non dovrebbe esser necessario sostituire la ROM degli A/500 perchè le pochissime modifiche riguardano più il Workbench che il Kickstart; questo è stato modificato essenzialmente in attesa dei nuovi chip custom in grado di indirizzare 1 Mega di memoria. Quando, comunque, tra un bel po' di tempo tale software sarà ufficialmente importato, sentiremo dalla Commodore come si farà per sostituire l'hardware di chi ha il Kickstart su ROM. Prevediamo, inoltre, che le versioni 1.2 e 1.3 convivranno per moltissimo tempo ancora.
• Molti programmi commerciali, sviluppati prima dell'avvento di A/2000 e A/500, non funzionano con l'espansione di memoria e, se non si puo "tagliare" la RAM supplementare col comando NoFastMem, non c'e nulla da fare (se non smontare, sui 500, l'espansione stessa!). Unico rimedio è comprare un'espansione non Commodore di tipo non-autoconfigurante.
• I videodigitalizzatori Digiview (o cloni italiani) funzionano al meglio con telecamere b/n perchè queste dispongono di una maggiore nitidezza; un'apposita elaborazione provvede a riprodurre immagini a colori ricorrendo al sistema della "somma cromatica". Per le connessioni da effettuare bisogna rivolgersi al rivenditore per sapere di quali segnali si può disporre all'uscita del videoregistratore e se questi sono compatibili col digitalizzatore che interessa.
• I programmi di pubblico dominio sono scarsamente distribuiti anche dagli importatori paralleli. L'elenco dei Commodore Point è aggiornato (e pubblicato sulla nostra rivista) mese per mese; per il momento, quindi, la maggior concentrazione dei C.P. è al Nord; tra breve verranno conclusi accordi commerciali per espandere punti di vendita anche al centro Sud (isole comprese, pranzo e cena al sacco, senza architetti).

AMIGA

MONITOR

□ Ho una confusione notevole per ciò che riguarda i monitor da collegare all'Amiga.
(Giovanni Badalamenti - Palermo)

• I monitor 1081 e 1084 hanno una risoluzione grafica intrinseca inferiore a quella gestibile con l'Amiga, ma è comunque superiore a quella di un comune TV color. Quel monitor di marca non Commodore, che citi nella lettera, è forse più economico, ma sicuramente meno valido dell'originale Commodore: tocca a te decidere se la differenza di prezzo è compatibile con la differenza di qualità.

conosciute (stessa cosa per If, Then :Else).

• La stampante che chiedi (economica, trasportabile, affidabile, rapida, silenziosa e per togli protocollo) devono ancora inventarla! Però non credere che siamo tanto lontani dal vederla in vetrina a poche lire. Una stampante del genere oggi costa parecchio, ma il progresso tecnologico offre, giorno dopo giorno, macchine sempre più precise, veloci e, soprattutto, economiche.

PROBLEMI CON I FILE

□ Utilizzando file su disco e su nastro mi capita un fenomeno per me inspiegabile (segue descrizione).
(Pierluigi Baglioni - Roma)

• Il comando Input è un po' particolare e ce ne possiamo accorgere perfino se lo usiamo quando non è riferito al drive, ma al video:

```
100 Input A$
110 Print A$
120 Goto 100
```

Se, in risposta al punto di domanda di riga 100, si batte Return (senza battere alcun altro tasto) oppure si preme la barra spaziatrice (prima di battere Return) si ottiene un effetto strano: riappare la stringa digitata prima oppure (se è la prima volta che battiamo i tasti dopo Run) una stringa nulla.

Un fenomeno analogo capita anche con i file sequenzaili (su disco o su nastro). In fase di scrittura (Print#) eventuali stringhe nulle, oppure costituite da soli caratteri di spazio, vengono effettivamente scritte come tali. Tuttavia, in fase di lettura, vengono gestite alla stessa stregua del comando Input relativo alla tastiera, cioè vengono ignorate! L'unico modo per "recuperare" ciascun carattere appartenente al file, è quello di ricorrere all'istruzione Get#, che presenta, però, lo svantaggio di una notevole lentezza di esecuzione.

Per aggirare l'ostacolo è necessario individuare, in fase di scrittura, le stringhe eventualmente nulle (o costituite da spazi) e sostituirle da caratteri-codice a piacere, come ad esempio alcuni asterischi, o altri caratteri che non diano, comunque, luogo ad equivoci in fase di successiva lettura.

SORT ULTRAVELOCE

□ In un mio programma ho la necessità di creare una matrice, spesso molto lunga, e di doverla ordinare alfabeticamente o numericamente. Il Sort Basic da me realizzato, però, richiede a volte alcune ore!
(Alessandro Marrazzo - Colleferro)

• Sul N. 31 di Commodore Computer Club è riportato un programma in l.m., di velocissima esecuzione, che aggiunge, al C/64, un comando Basic in grado di effettuare l'operazione richiesta di Sort (=ordinamento). Purtroppo, però, può essere utilizzato solo con un vettore (monodimensionale) e non può essere applicato a matrici (pluridimensionali).

Tuttavia può esser possibile "spaccare" una matrice bidimensionale in due vettori e realizzare, a parte, un vettore indice (numerico) in cui ad ogni elemento corrisponda il relativo elemento alfanumerico. In seguito si potrebbero ordinare i vari vettori, ad alta velocità, ed "estrarne" gli indici di riferimento.

AMIGA

LIBRI AMIGA

□ Che libri consigliate agli utenti Amiga?
(Vittorio Bagnato)

• Parleremo dei libri in italiano proposti per Amiga, a patto che siano davvero interessanti. Riteniamo, comunque, che servano solo per muovere i primissimi passi; in seguito bisognerà arrangiarsi da soli ed imparare tutte quelle cose "complementari" e pratiche che i libri non possono spiegare, ma che provvediamo a descrivere su queste pagine.

Per ciò che riguarda i vari volumi attualmente in libreria, stai attento alle fregature: spesso non sono altro che adattamenti del libretto di istruzioni che già possiedi, oppure trattano argomenti di scarsissimo interesse.

Prima di acquistarli, sfogliali con attenzione in modo da esser sicuro della loro pratica utilità.

risposte rapide

POKE C/16
(Fulvio Collodi)

Le Poke per C/16 che hai inviato mi pare di averle già pubblicate, comunque le riporto nuovamente:
Le locazioni 2021, 2022, 2023, 2024 fissano i margini della finestra impostata. Dopo aver assegnato i valori desiderati è necessario impartire Sys 52886. Altre Sys inviate dal nostro lettore: 42972 (save); 42992 (verify); 42993 (load).

NON PER VIC/20
(Matteo Bruno - Milano)

Il programma "Sistemi bi-ridotti" gira su qualunque computer, tranne che sul Vic 20 a causa della sua limitata disponibilità di memoria Ram. Ma che aspetti a comprare un C/64?

SPRITE C/128
(Stefano Frattari - Civitanova)

Sul n. 51 (marzo scorso) è stato pubblicato un lungo articolo ("Muoversi in fretta tra gli sprite") che spiega in modo piuttosto dettagliato la corretta gestione degli sprite nel C/128.

KOALA SU CASSETTA
(Gaetano Torcetta - Firenze)

Tutti i listati che elaborano le immagini grafiche gestite con il programma grafico Koala per C/64 (versione cassetta oppure disco) possono essere utilizzati servendosi del semplice registratore a cassette.

SCACCHI
(Luigi Cirillo - Trento)

I programmi di scacchi (come tutti quelli commercializzati) bisogna accettarli così come sono: non è assolutamente possibile apportare modifiche. Penso, però, che con il C/64 tutto ciò che era possibile realizzare in ambito scacchistico sia già stato attuato. A meno di esser smentito dalla solita, imprevedibile software house che, magari, provvederà a fornare un simulatore di Kasparov.

COMPLIMENTI
(Giorgio Buttazzone - Milano)

Ti ringrazio delle gentili parole che hai voluto rivolgermi.

ROUTINE DI TOMA
(Luca Vignale - Brandizzo)

Le routine grafiche di Toma, di cui spesso parliamo, sono routine in linguaggio macchina che un nostro collaboratore (Danilo Toma, appunto), ha sviluppato. Aggiungono alcune istruzioni grafiche a quelle standard del C/64 e consentono di disegnare in prospettiva con la massima facilità. Te le puoi procurare su disco richiedendo anche (ti conviene) il fascicolo "Commodore speciale" appositamente dedicato. Telefona al nostro servizio arretrati per avere maggiori informazioni, oppure riferisciti alla nostra pubblicità (Commodore speciale: Linguaggio Macchina e Routine grafiche).

LOGO
(Giovanni Lenci - Cappella M.re)

Per ciò che riguarda il linguaggio Logo (originale) è necessario riferirsi al libretto di istruzioni; se la funzione richiesta non è prevista, purtroppo, non c'è nulla da fare.

MODEM
(Giacomo Longo - Messina)

Non è agevole spiegare in poche righe a che cosa possa servire un modem. Posso dire che, tuttavia, il 6499, da applicare al C/64, è uno strumento semplice, economico e certamente idoneo per fare i primi passi nel mondo della telematica. Attenzione, però, alla bolletta del telefono!

LETTERE
(Giovanni Ettorre - Guardia)

Un validissimo programma di word processor (per scrivere lettere e documenti in generale) è certamente l'Easy Script per C/64; il suo prezzo è molto contenuto, soprattutto se si tiene conto che le istruzioni accluse sono in italiano. Per procurarselo, rivolgersi ai Commodore Point, oppure alle ditte le cui inserzioni pubblicitarie compaiono sulle nostre pagine.

TUTTO PER IL TUO COMMODORE

La Delta Computing, di Firenze, ha deciso di offrire i propri servizi non solo ai grossisti ed ai riparatori, ma anche agli utenti finali dei prodotti Commodore.

Chi, dunque, possiede un C/64, C/128, C/16, Plus/4 ed un qualsiasi modello di Amiga (per non parlare di 1541, 1571, mps 801/2/3), potrà, d'ora in poi, rivolgersi alla Delta Computing, che continua, tuttavia, la sua attività di distributore ufficiale della Commodore Semiconductors Group per l'Italia.

Da Aprile, oltre alla distribuzione dei circuiti integrati originali Commodore, inizierà la commercializzazione dei prodotti Rex Datentechnik, azienda leader, in Germania, di sofisticati accessori di alta tecnologia per computer (schede elettroniche, Ram aggiuntive, programmatori di Eprom, cartucce, eccetera).

Non appena avremo modo di provare questi apparecchi, che sembrano decisamente interessanti, ne riparleremo certamente.

Come se non bastasse, all'hobbysta, come al professionista, saranno proposti anche circuiti elettronici in Kit di montaggio, particolarmente idonei per coloro che intendano avvicinarsi per la prima volta all'elettronica pratica.

Per ulteriori informazioni:
Delta Computing
Via A. Bertani, 24
50137 Firenze
Tel. 055/609.227 - 608.803

VIVERE CON UN PC/AT

Una breve chiacchierata informale sulle differenze esistenti tra il pianeta C/64 e quello dei PC/XT/AT e compatibili vari

L'utente di un home computer, per ciò che riguarda la sua semplicità d'uso, è il più fortunato essere vivente appartenente alla fauna informatica.

Non deve fare altro, infatti, che collegare l'uscita video al suo TV, accendere il computer e caricare uno dei giochini ricevuti in omaggio dal rivenditore per smanettare con il joystick.

Un home computer, non dimentichiamolo, "deve" avere le caratteristiche della semplicità e dell'immediatezza d'uso: sono anzi queste le peculiarità che ne hanno permesso una diffusione così capillare tra strati della popolazione non molto esperte di misteri dell'elettronica.

Il C/64 (e, prima di lui, il Vic/20) aveva, inoltre, la caratteristica della completezza: imbottito di Ram fin dalla configurazione standard, dotato di circuito video e sonoro ottimali, offriva ben due prese per il joystick all'interno delle quali erano addirittura presenti i sensori per l'eventuale penna ottica e per le paddle. L'interfaccia seriale e la porta utente erano di serie e non attendevano altro che di esser collegate ai vari apparecchietti che, puntualmente, sono comparsi sul mercato in seguito allo

straordinario successo commerciale del C/64.

L'espandibilità, quindi, non era una caratteristica di questo computer perchè, in effetti, era già espanso al massimo delle sue possibilità; ciò non impedì, tuttavia, di presentare, in seguito, espansioni di memoria, dal successo più o meno accentuato.

Una faccenda diversa

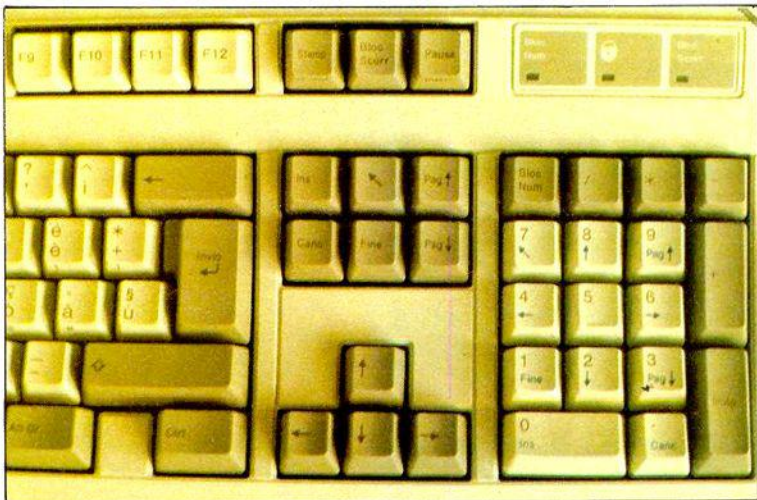
Ben altra origine (e destinazione) hanno avuto altri computer che fondavano la propria validità sull'elevato grado di espandibilità vantato.

Senza parlare dell'Apple, che meriterebbe un discorso a parte, ci intratterremo sulle decisoni della IBM, che presentò l'ormai storico PC, visto ufficialmente per la prima volta in Italia nel lontano 1983.

Il primo modello evidenziava la miopia tecnica e commerciale dei progettisti IBM, che sarebbe costata, negli anni successivi, la perdita di grosse fette di mercato a favore di marche con fatturato minore (tra cui la Commodore stessa).

Ai tempi della presentazione del PC, infatti, erano già disponibili modelli di computer le cui caratteristiche risultavano molto apprezzate dagli utenti e facevano prevedere, in modo indiscutibile, la decisione degli utenti stessi di privilegiare, negli anni a venire, elaboratori dotati di caratteristiche ben definite, ormai irrinunciabili.

Il PC IBM, invece, si presentava in bianco e nero (il colore non era nemmeno previsto), con un penoso "apparato" sonoro (in realtà un semplice cicalino), una gestione dei dischi arcaica e dotata di capacità e velocità praticamente identiche, se non inferiori, ai computer "personali" già in commercio.

L'unico motivo per cui il PC IBM ebbe successo, a nostro modesto avviso, fu il nome che portava: se, invece di IBM, il nome fosse stato "Pippo Computer" avrebbe fatto una brutta fine perchè, dal punto di vista tecnologico, era dotato di contenuto modesto.

Se, però, nel campo cosiddetto "professionale" la gente badava più al nome che ad altro, nel campo hobbistico l'utente medio non era altrettanto sprovveduto; un qualsiasi ragazzino americano degli anni '80 sapeva benissimo le differenze tecniche tra un modello e l'altro, ne intuiva i possibili sviluppi e capiva che il C/64 aveva un futuro assicurato. In conseguenza di ciò, il timido tentativo IBM di proporre il PC junior (una sorta di computericchio al di fuori di qualsiasi realtà tecnologica e commerciale) fallì miseramente.

Forse fu proprio questo insuccesso che fece capire alla IBM che, per vendere, non bastava più solo il nome, ma era necessario mettere nella confezione un apparecchio che potesse reggere il confronto, senza sfigurare, con altri elaboratori concorrenti.

Se non bastasse, poi, si spargeva la notizia che non solo alcuni produttori erano in grado di copiare il PC ma, addirittura, di realizzare modelli migliori.

Nonostante il maquillage dei successivi apparecchi, alla IBM sfuggì la considerazione che le aggressive fabbriche orientali erano in grado di simulare in modo perfetto le funzionalità dei suoi prodotti, a dispetto di circuiti speciali (il famoso BIOS) superprotetti legalmente. Nella quasi totalità dei casi, infatti, le copie dei PC facevano girare i programmi professionali (anch'essi facilmente pirateggiati) ad una velocità maggiore che nel PC "originale". Una vecchia battuta che circolava a quei tempi, diceva che il meno compatibile, tra i computer compatibili, era proprio il PC IBM.

Il colore, quasi disprezzato agli inizi, fece la sua comparsa dapprima con la disponibilità di soli 4 colori, poi, finalmente, si decisero a gestire 16 colori quando, però, i computer concorrenti erano già in grado di offrire decine di colori diversi.

Il PC, nonostante tutto

Nonostante tutto, però, lo standard lanciato da IBM teneva duro e dettava legge a tutti, clonatori taiwanesi in testa. Non possiamo dimenticare, infatti, che nella moderna società industriale il "nome" fa ancora molto effetto ed ancor oggi IBM è sinonimo di computer di alto livello.

Ciò giustifica il fatto che, a dispetto della arcaicità intrinseca del sistema Ms/Dos, lo standard più utilizzato negli uffici, e nelle applicazioni professionali in genere, sia ancora quello imposto da Big Blue; qualcosa,

però, sta cambiando e lo dimostra il fatto che, anno dopo anno, la IBM si adegua alle richieste del mercato, è "costretta" ad osservare l'attività della concorrenza ed a proporre allineamenti tecnologici per non perdere consistenti fette di mercato.

Dopo aver proposto i computer della serie PC, poi quelli XT e, dotati di microporocessore 80286, gli AT, la IBM si illude ancora, nonostante l'evidenza, di essere in grado di fabbricare computer incopiabili. Già si vedono, soprattutto in oriente, schede, computer ed aggeggi elettronici garantiti come totalmente compatibili con il nuovo standard; la possibilità di clonazione, d'altra parte, rappresenta una valida affermazione dello standard che, se realmente incopiabile, non avrebbe riscosso il successo che, invece, sembra ottenere.

Le difficoltà

Tenendo conto di quanto detto, pertanto, l'utente finale, specialmente inesperto, paga pesantemente le "sviste" tecnologiche e l'aggiornamento continuo cui è stato sottoposto il PC nel corso degli anni.

L'adattabilità più immediata (e, fortunatamente, semplice) è quella relativa al sistema operativo Ms-Dos la cui versione più popolare è, attualmente, la 3.1. Per una serie di motivi, tra cui l'esigenza di compatibilità con le versioni precedenti, il sistema operativo consente di assegnare ai file nomi con lunghezza massima di soli otto caratteri (con estensione facoltativa di tre) provocando non pochi disagi a chi considerava "stretti" i 16 caratteri del C/64. L'eventualità, per nulla remota, di cancellare accidentalmente alcuni file (registrandone di nuovi, dotati dello stesso nome) è ancora presente e non se ne capisce il motivo.

SIMONE GALLINA

Molti sono, però, i comandi standard aggiunti alle precedenti versioni, ed ottimi e completi, per fortuna, i numerosi programmi di utilità che consentono di impartire i comandi del DOS senza sbagliarne la sintassi.

Uno scoglio per nulla gradito è, invece, la questione delle schede grafiche, che non poche delusioni ha suscitato negli utenti per i quali la gestione della grafica in Hi-Res (magari a colori) era una caratteristica scontata.

Le delusioni, dicevamo, vengono subito dal monitor, acquistato insieme al PC.

Di solito, per non aumenatare il prezzo finale di vendita, il monitor proposto è un semplice b/n. Questo fatto induce spesso l'utente a ritenere che, facendo girare programmi "colorati", li possa utilizzare egualmente, pur se visualizzando schermate con vari toni di grigio (o di verde, a seconda dei montor), come era abituato con il C/64 collegato ad un TV in bianco e nero.

Ma ecco il primo inconveniente dello standard Ms-Dos: un programma scritto per girare con una scheda a colori non può girare in bianco e nero a causa della totale diversità di gestione delle due configurazioni. Per fortuna quasi tutti i programmi professionali (cioè importanti) si configurano automaticamente a seconda della scheda montata oppure, tramite un menu, chiedono di indicare il tipo di scheda e di monitor posseduti.

Nei casi in cui, invece, è tassativamente richiesta la scheda a colori ed il monitor specifico, non è possibile far girare il programma.

Tra le schede, se non bastasse, c'è una confusione sconosciuta agli utenti di C/64: vi è la Hercules monocromatica e quella a colori, c'è la

CGA (4 colori) la EGA (16 colori) le varie Superega (che offrono altri colori ed altre risoluzioni) oltre ad altre schede che offrono peculiarità più o meno sfruttabili in package professionali.

Il fatto negativo è che, della carenza della scheda grafica, ci si accorge solo quando, caricato un programma, questo non gira!

Molti calcolatori, tuttavia, montano speciali schede e sono forniti di software opportuno che consente di visualizzare, su di un modesto monitor b/n, programmi che richiederebbero, al contrario, solo schede e monitor a colori.

La configurazione della tastiera è semplicissima (basta caricare il file specifico prima di operare); di diversa portata, invece, la "distrazione" dovuta alla mancanza di memoria aggiuntiva.

Molti programmi, infatti, richiedono ormai configurazioni minime di memoria Ram che coincidono molto spesso con la configurazione... massima: chi ha acquistato un XT compatibile con soli 256 KRam si accorge quasi subito che deve espandere il suo sistema.

Non si fa a tempo, però, ad espandere la Ram, che subito ci si accorge, con raccapriccio... finanziario, che i programmi più interessanti richiedono due disk drive e l'hard disk: è praticamente impossibile "fare" del word processor con un solo floppy mentre è negata la possibilità di lavorare con un package integrato (per non parlar di Desk Top Publishing) in assenza di hard disk.

La disperazione sembra invadere l'animo del neo utente quando (dulcis in fundo) non può fare a meno di notare che il mouse è molto comodo per gestire i programmi professionali: ma quale mouse, fra i tanti presenti sul mercato?

Documentarsi prima di un acquisto è, in conclusione, un'operazione doverosa per evitare cocenti delusioni.

Coraggio!

Sembrerebbe, dalle note di queste pagine, che un sistema Ms-Dos, in generale, sia da scartare a priori; naturalmente la situazione non è affatto così nera come potrebbe apparire a prima vista: anzi, vi posso assicurare che, da quando possiedo il mio AT compatibile, l'orizzonte dell'informatica mi appare ancora più vasto, e determinate applicazioni s/w le svolgo con enormi soddisfazioni, anche grazie alla pregevole fattura dei package disponibili, realmente professionali, affidabili e privi di bug.

L'importante, volendo passare ad un sistema "superiore", è non considerare un computer professionale come se fosse un C/64 più grosso, magari più semplice da usare. Ciò può esser vero, ma solo a patto di tener ben presenti i punti che seguono:

- Un computer Ms-Dos è nato per scopi professionali: l'aspetto ludico, pur se possibile (e presente), non è quello privilegiato dalle software house. Scordatevi, insomma, gli effetti sonori del C/64 ed i vari giochini di un certo tipo.
- Se ne avete la possibilità, procuratevi subito la configurazione espansa a 640 KRam; tanto, prima o poi, dovrete comperarla lo stesso, vi piaccia o no.
- Tra i vari acquisti prevedete di procurarvi, entro breve tempo, la scheda grafica (meglio se una EGA, anche se compatibile, ormai economica) e relativo monitor a colori, se non lo possedete.
- Volendo operare con programmi prevalentemente professionali, magari lasciate perdere la scheda a colori, ma acquistate subito un hard disk affidabile.
- Il processore matematico (8087 oppure 80287) non è strettamente indispensabile soprattutto perchè pochi programmi ne utilizzano le risorse.
- Procuratevi, magari più in là nel tempo, un mouse ed accertatevi che i package che usate di solito lo "riconoscano".
- Non illudetevi di avere a disposizione un "vero" Ms-Dos con una cifra paragonabile a quella di un sistema C/64 o C/128: a volte si vede la pubblicità di apparecchi Ms-Dos compatibili supereconomici che, però, sono sforniti di monitor, memoria, disk drive, interfacce ed altre "cose" piuttosto preziose per un utilizzo serio di un computer.

UN'INSOLITA RAM-DISK PER IL C/128

Se non si usa il potente computer nella modalità 80 colonne, questa routine può evitare frequenti accessi al drive ed al registratore

di **Mario Beccia**

Il semplice programma proposto consente di implementare nella memoria del C-128 una pseudo ram-disk, un'area, cioè, dove un programma può essere registrato temporaneamente, fino allo spegnimento della macchina oppure fino ad un reset di sistema.

La lunghezza massima del programma immagazzinabile è di ben 16K Ram. La routine è allocata da $1300 a $1376, ma risulta completamente rilocabile in qualsiasi altra parte della memoria a disposizione.

La novità peculiare della routine pubblicata consiste nel fatto che essa non sottrae neppure un byte all'area del Basic, poichè utilizza, come buffer (=magazzino temporaneo), i 16K Ram interni al VDC 8563, vale a dire il circuito integrato che gestisce la visualizzazione nel formato 80 colonne.

Proprio per questo motivo non è possibile passare nella modalità 80 colonne mentre è presente un programma nel buffer, pena la visualizzazione di caratteri casuali e la contemporanea cancellazione del programma ivi presente.

La routine l.m. trasferisce 16k dal banco zero al buffer, a partire dall'indirizzo $1C00 fino a $5C00. Malgrado utilizzi la modalità fast ed il blanking, non risulta una routine molto veloce (si fa per dire), a causa dello strano metodo di accesso alla RAM del VDC, imposto dal chip stesso.

La routine utilizza due puntatori in pagina zero ($FD-$FE): attenzione, quindi, ad evitare sovrapposizioni. Per guadagnare velocità si è ritenuto necessario bloccare le IRQ durante le operazioni di caricamento e salvataggio; ne consegue che, ricorrendo spesso alle funzioni della routine, l'orologio interno TI$ non è più attendibile.

Il programma utilizza alcune subroutine del Sistema Operativo: la prima è posta a partire da $CDCA e l'altra da $CDD8, che, in pratica, sono le trasposizioni delle istruzioni PEEK e POKE nella RAM dell'8563.

Bisogna infatti sapere che, per accedere alla suddetta RAM, occorre impostare due puntatori, compresi nel suo interno, all'indirizzo desiderato, scomposto in byte basso e alto.

Per ottenere lo scopo prefissato si deve usare un metodo piuttosto insolito: bisogna POKare in $D600 il numero del puntatore ($12-$13) ed in $D601 il valore da inserire in esso. Una volta impostati i due puntatori, non resta che leggere il valore del byte puntato, leggendo la locazione $1F, sempre seguendo il procedimento prima esposto.

Il metodo descritto ha un notevole pregio: dopo aver impostato $12-$13 alla locazione desiderata, non dovremo più curarci di loro poichè si incrementeranno automaticamente ad ogni lettura.

NEWEL srl

hardware software telematica

20155 MILANO - Via Mac Mahon, 75
tel. 02/32.34.92 - tel. 02/32.70.226

NEGOZIO AL PUBBLICO
E VENDITA PER CORRISPONDENZA

CASH & CARRY

COMMODORE POINT '88

COMMODORE AMIGA 500/2000

"AL PREZZO PIÙ BASSO D'ITALIA" »CON GARANZIA & OMAGGIO«

Amiga VID — L. 139.000
Digitalizzatore di immagini per Amiga 500/1000/200

Amiga Syntetic — L. 170.000
Digitalizzatore Audio per Amiga 500/1000/2000

Amiga VIDEOSOUND — L. 290.000
Digitalizzatore Audio VIDEO, tutto in uno.
Ottimo, per 500/1000

INT MIDI AMIGA PLUS — L. 79.000
Nuova interfaccia midi per Amiga 500/1000/2000

DRIVE AGGIUNTIVO AMIGA 500/1000 (SLIM LINE
L. 235.000

ESPANSIONE 512 PER AMIGA 500
INT. con orologio — **L. 199.000**

ESPANSIONE 2MB per A500/1000
L. 850.000

EMULATORE 64 per AMIGA — L. 49.000
L'UNICO EMULATORE VERAMENTE FUNZIONANTE CON L'AUDIO E CON LA POSSIBILITÀ DI SALVARE I FILES SU 3 1/2

VID-AMIGA II — L. 899.000
Novità digitalizzatore in tempo reale per A 500/1000/2000

NOVITÀ **PENNA OTTICA PER AMIGA 500/1000/2000** — L. 149.000

NOVITÀ **DRIVE 5 1/4 per amiga** — L. 299.000
Permette di CARICARE pvg. del PC-IBM

DRIVE INTERNO PER AMIGA 2000 (MECC. NEC)
L. 199.000

NOVITÀ **VIDEON (AMIGA)** — L. 290.000
Nuovo digitalizzatore di immagini - Lavora senza filtri e in "PIÙ direttamente da Videoregistratore o Telecamera!

OLTRE 1000 PROGRAMMI AMIGA
RICHIEDERE CATALOGO

COMMODORE 64/128 - COMMODORE 64/128

THE NEW FINAL TURBO III per 64/128 (modo 64) L. 69.000
L'emulazione continua!!!
Eccovi l'ultima release della mitica cartuccia notevolmente migliorata e modificata. Turbola favolosa routine dello speddos su cartuccia fino a 10 volte più veloce sia in lettura che in scrittura!!! 8 tasti funzione programmati. 24 K ram per i grog. in Basic. Un favoloso protettore di programmi tipo O.M.A. incorporati. Dischi e cassette IN UN SOLO FILE!!! (boot se necessita. Inoltre ha incorporato il GAME KILLER (evita la collisione degli sprite ed ha ben 40 comandi Basic Turbo a disposizione ... HARDCOPY "HP". Premendo un solo tasto potrete fare l'ardcopy del video in 12 gradazioni di grigio.
ECCEZIONALE!!!

EPRON NEW GRAPHIC MPS 803
Si sotituisce il generatore di caratteri della stampante MPS-80 (per migliorare la leggibilità della scrittura con quattro nuovi set di caratteri.
L. 49.000

PROCESSORE VOCALE
L. 115.000
Digitalizzatore vocale tipo "Voice Master" notevolmente migliorato composto a cartuccia hardware e microfono software interamente in italiano con ampio manuale di istruzioni. Incredibile, fa parlare il tuo Commodore 64 puoi programmarlo a fin che conosca la tua voce e ti risponda.

KIT PULIZIA DRIVE
5 1/4 o 3 1/2
L. 15.000 cad.

DISPONIBILI TUTTI I PEZZI DI RICAMBIO CCOMMODORE 64
Sconti particolari per rivenditori e quantitativi.
TELEFONATE! Per ulteriori informazioni richiedete i cataloghi per il vostro Computer il settore, inviando L. 1.000 in francobolli. Ricorda che alla **NEWEL**trovi anche tutto per il COMMODORE AMIGA 64-128 MSX, SINCLAIR ZX & QL ATARI ST e PC compatibili.

NUOVA GUIDA AL CP/M 128 IN ITALIANO
L. 30.000

Alimentatori professionali
- - - -
per 64 — L. 39.000
per 128 — L. 69.000

VIDEODIGITAL 64
Nuovo digitalizzatore in cartuccia, digitalizza le tue più belle immagini con l'aiuto di una videocamera o videoregistratore semplicissimo da usare con manuale in italiano. Inoltre è possibile modificare le immagini con il KOALA ecc.
L. 70.000

STARDOS NEW! Eccezionale novità un velocizzatore che supera persino la velocità dello speed-dos attiva i tasti funzione ecc. In una sola Eprom kit da inserirsi nel C64 con manuale in ital. Non necessita di elaborazioni al drive ne del cavo parallelo.
L. 39.000

Adattatore telematico 64 Commodore V21-V23
TUTTO IN ITALIANO — L. 99.000

REALTIME DIGITAL 64 — L. 170.000
Notevolmente migliorato con possibilità di animazioni e videoclip.
IL PRIMO DIGITALIZZATORE
IN TEMPO REALE PER IL TUO 64/128

DRIVE OC-118 per 64/128
Drive Slim-line 100% compatibile
a sole L. 249.000

NOVITÀ **The CARTRIDGE (NOVITÀ) L. 79.000**
Nuova cartuccia multiutility con:
Turbo, Superturbo fino a 10 volte più veloce,
Sprite Killer, Poker, Monitor
Utility e comandi aggiuntivi, Toolkit,
tasti funzione, copyfiles, ecc.
PIÙ UN ECCEZIONALE SPROTETTORE CASSETTA – DISCO E VICEVERSA IL TUTTO IN UN UNICO FILE
TUTTO IN ITALIANO !!!
Compatibile speed-dos e drive compatibili!

NOVITÀ **NOVITÀ ASSOLUTA!!!**
The ICONE CARTRIDGE
per 64/128
L. 90.000
Un nuovo sistema operativo a ICON e finestre grafiche che racchiudono le caratteristiche della final III & The cartridge 64 e molto più!! Potrete valorare quasi come con il Geos 64 (MAN. IN ITALIANO)

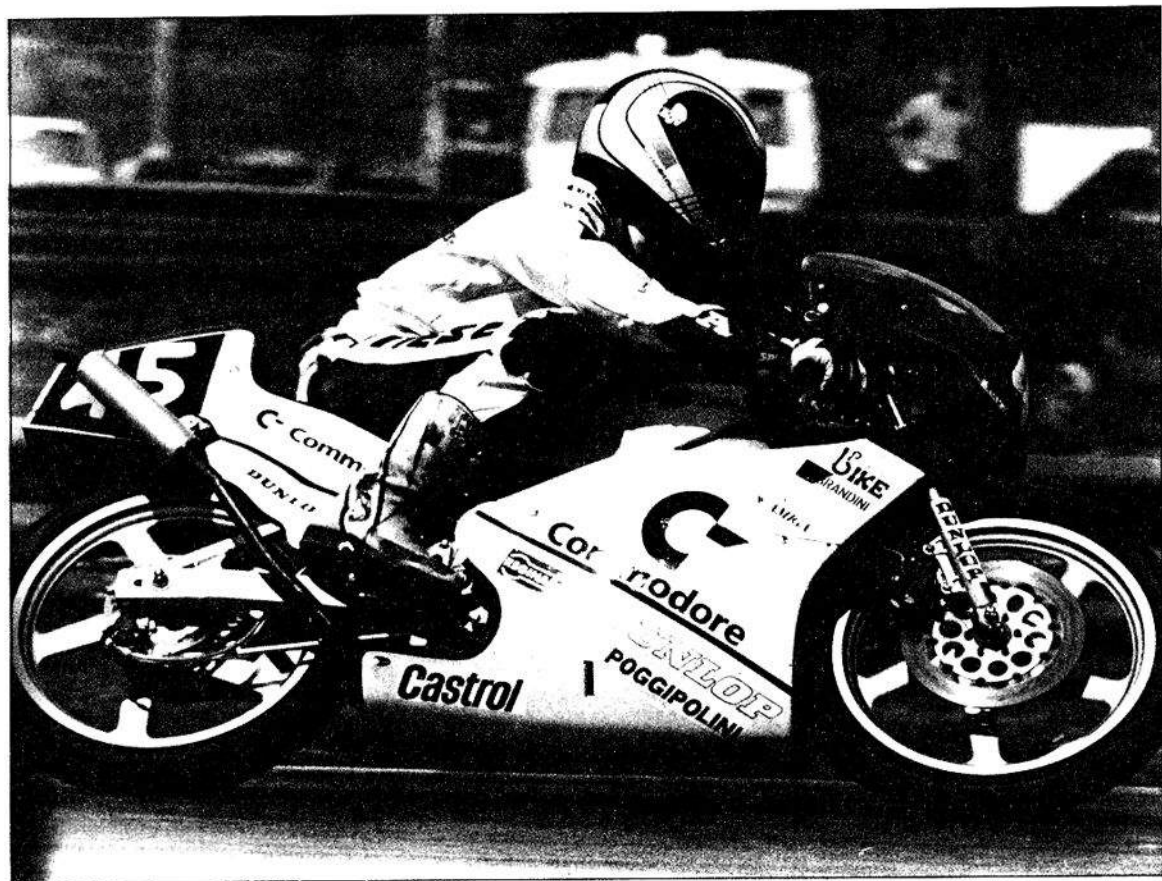
IN OFFERTA

O.M.A. PLUS	L. 60.000
TURBO FREEZE	L. 50.000
SPEED-DOS 1541	L. 49.000
SPEED-DOS 1541/C	L. 59.000
PENNA OTTICA	L. 49.000
TAST LOAD - RESET	L. 30.000
KIT EPROM 80	L. 25.000
KIT GRAFICO 802	L. 35.000
RESET 64	L. 10.000
COPRITASTIERA 64	L. 10.000
COPRITASTIERA 64 NEW	L. 15.000
COPRITASTIERA 128	L. 18.000

CHIUSURA DAL 25 LUGLIO AL 10 SETTEMBRE PER RINNOVO LOCALI

```
100 REM* RAM-DISK   PER C/128 *
110 REM*  IN MODO 40 COLONNE  *
120 REM** NOTE PER L'USO : ****
130 REM SYS 4864,0 PER SAVE   *
140 REM SYS 4864,1 PER LOAD   *
150 REM* BY M. BECCIA -  ROMA *
160 :
170 SA=4864:C=6:B=1:FAST:CK=0
180 DO:READ A:IF A<0 THEN EXIT
190 POKE SA,A:SA=SA+1
200 COLOR4,C:C=C+B:B=-B
210 CK=CK+A
220 LOOP
230 :
240 :
250 COLOR0,1:COLOR4,1
260 READ CG:SLOW:SCNCLR
270 PRINT"[illegible]"TAB(5);
280 IF CG<>CK THEN 500
290 END
300 :
310 DATA 120,72,169,0,141,0,255,169
320 DATA 253,141,48,208,169,0,141,17
330 DATA 208,162,18,169,0,32,204,205
340 DATA 232,32,204,205,169,0,133,253
350 DATA 169,28,133,254,160,0,104,240
360 DATA 29,162,0,142,0,255,32,216
370 DATA 205,162,63,142,0,255,145,253
380 DATA 200,208,236,230,254
390 DATA 165,254,201
400 DATA 92,208,228,24,144,26,162,63
410 DATA 142,0,255,177,253,162,0,142
420 DATA 0,255,32,202,205,200,208,238
430 DATA 230,254,165,254,201
440 DATA 92,208,230
450 DATA 169,0,141,0,255,169,252,141
460 DATA 48,208,169,27,141,17,208,169
470 DATA 63,141,0,255,88,96
480 DATA -1,16835
490 :
500 PRINT"ERRORE NEI DATA":END
```

```
;  Inizializzazzione generale
;
. 01300  78        sei ; Blocca le IRQ
. 01301  48        pha ; Posizaiona A
                       ; sulla pila
. 01302  a9 00     lda #$00  ; Bank 15
. 01304  8d 00 ff  sta $ff00 ;
. 01307  a9 fd     lda #$fd  ; Fast mode
. 01309  8d 30 d0  sta $d030 ;
. 0130c  a9 00     lda #$00  ; Attiva il
. 0130e  8d 11 d0  sta $d011 ; blanking  ;
; Setta i puntatori usati dal programma ;
. 01311  a2 12     ldx #$12
. 01313  a9 00     lda #$00
. 01315  20 cc cd  jsr $cdcc
. 01318  e8        inx
. 01319  20 cc cd  jsr $cdcc
. 0131c  a9 00     lda #$00
. 0131e  85 fd     sta $fd
. 01320  a9 1c     lda #$1c
. 01322  85 fe     sta $fe
. 01324  a0 00     ldy #$00
. 01326  68        pla        ;sceglie tra
. 01327  f0 1d     beq $1346 ;load e save;

; Ciclo principale per 'load'
;
. 01329  a2 00     ldx #$00
. 0132b  8e 00 ff  stx $ff00
. 0132e  20 d8 cd  jsr $cdd8
. 01331  a2 3f     ldx #$3f
. 01333  8e 00 ff  stx $ff00
. 01336  91 fd     sta ($fd),y
. 01338  c8        iny
. 01339  d0 ec     bne $1327
. 0133b  e6 fe     inc $fe
. 0133d  a5 fe     lda $fe
. 0133f  c9 5c     cmp #$5c
. 01341  d0 e4     bne $1327
. 01343  18        clc
. 01344  90 1a     bcc $1360
;
;  Ciclo principale per 'save'
;
. 01346  a2 3f     ldx #$3f
. 01348  8e 00 ff  stx $ff00
. 0134b  b1 fd     lda ($fd),y
. 0134d  a2 00     ldx #$00
. 0134f  8e 00 ff  stx $ff00
. 01352  20 ca cd  jsr $cdca
. 01355  c8        iny
. 01356  d0 ee     bne $1346
. 01358  e6 fe     inc $fe
. 0135a  a5 fe     lda $fe
. 0135c  c9 5c     cmp #$5c
. 0135e  d0 e6     bne $1346
;
;  Riporta alle condizioni iniziali.
;
. 01360  a9 00     lda #$00  ;Bank 15
. 01362  8d 00 ff  sta $ff00 ;
. 01365  a9 fc     lda #$fc  ;Slow mode
. 01367  8d 30 d0  sta $d030 ;
. 0136a  a9 1b     lda #$1b  ;Disattiva
. 0136c  8d 11 d0  sta $d011 ;il blanking
. 0136f  a9 3f     lda #$3f  ; Bank 0
. 01371  8d 00 ff  sta $ff00 ;
. 01374  58        cli   ;riattiva le IRQ
. 01375  60        rts
```

Attenti alla garanzia

A volte si creano spiacevoli equivoci sulla garanzia allegata alle macchine Commodore.

C'è da dire, anzitutto, che il foglio di garanzia è inserito in una busta sigillata che, a sua volta, è incollata esternamente alla confezione: non è possibile estrarla senza strappare irrimediabilmente la busta stessa. Morale della favola: rifiutate di acquistare macchine la cui confezione è priva della busta (ovviamente integra) ben visibile sulla confezione.

La garanzia stessa, inoltre, è composta da quattro parti: la prima è il tagliando di accettazione, da staccare ed inviare subito dopo l'acquisto.

La seconda è il vero e proprio certificato di garanzia, da conservare gelosamente in caso di bisogna.

La terza è formata da tre tagliandi di manutenzione sui quali è presente (come sulle altre due parti) il numero di matricola della macchina acquistata; è quindi cura dell'acquirente verificare subito la corrispondenza del numero di matricola sull'apparecchiatura, sul certificato e sulla confezione.

All'interno della busta, inoltre, dovrebbero esser presenti l'elenco dei centri di assistenza e la cedola di abbonamento alla nostra rivista.

Commodore in fiera

Confermata la presenza della Commodore sia al SIM (8 - 12 settembre) che allo SMAU (29 settembre - 3 ottobre) di Milano.

Inutile dire che la superficie occupata sarà più ampia di quella utilizzata nelle manifestazioni precedenti: maggiori dettagli sul prossimo numero di C.C.C.

Potremo ammirare da vicino anche la favolosa Honda 125 sponsorizzata Commodore che, guidata da Massimo Bonfante, è attualmente (23 maggio) prima in classifica, con 30 punti, nella classe 125 monocilindriche.

Sempre di più

Continua con successo la "penetrazione" del mercato da parte della Commodore Italiana: oltre ai ben noti Commodore Point verranno tra breve resi operanti altri centri di vendita autorizzati, presso i quali si svolgeranno attività promozionali di particolare rilievo.

Degna di nota è la decisione di arricchire il Sud Italia di tali centri autorizzati.

Anche nel campo professionale la nuova, aggressiva politica commerciale sta mettendo a punto una catena di distribuzione ed assistenza adeguate allo sviluppo di tale significativo settore.

L'obiettivo finale è, come avevamo anticipato nei numeri scorsi, quello di bissare il successo conseguito in Germania, ove la Commodore ha venduto più personal computer delle altre marche.

Commodore Computer Center

Verrà potenziata l'attività dei centri Commodore, ben noti ai clienti più affezionati, presso i quali verranno svolti corsi di formazione aperti a tutti gli interessati. I temi trattati spaziano in tutti i campi dell'informatica: corsi di Basic per principianti, uso di package specifici (professionali e non); presentazione di nuove macchine e così via.

I centri dispongono di aule attrezzate con C/64, Amiga, Pc e le durate dei corsi variano dalla mezza giornata ai 2-3 giorni.

Per maggiori informazioni contattare i centri stessi, qui di seguito riportati in ordine alfabetico:

Alessandria Bitmicro Tel. 0131/44.32.52 • Bergamo Arcipelago Tel. 035/23.95.65 • Bologna Magicbus Tel. 051/27.07.1 • Catania Azeta Tel. 095/50.17.87 • Ferrara Mazzacurati Tel. 0532/47.00.8 • Livorno Etabeta Tel. 0586/88.67.67 • Modena Videodigit Tel. 059/23.92.66 • Napoli Tecnodata Tel. 081/77.41.21.3 • Pisa Electronic Service Tel. 050/98.22.02 • Ragusa Computershop Tel. 0932/20.10.0 • Reggio Emilia Computerline Tel. 0522/32.67.9 • Rimini Numerosoft Tel. 0541/27.10.9 • Roma Cenfor Tel. 06/47.43.09 • Roma Pixcomputer Tel. 06/82.93.50.7

Un transputer per Amiga

Amiga sarà il primo computer a basso costo a supportare una scheda speciale in grado di sperimentare la tecnologia dei transputer.

Il transputer, in parole povere, è quella struttura elettronica in cui una pluralità di processori si divide il compito di elaborare un unico programma complesso. Il risultato di un simile stratagemma consente di raggiungere velocità operative impossibili da conseguire con metodi tradizionali. Una delle applicazioni prevalenti sarà quella relativa al trattamento di immagini, ed alla grafica in generale, in cui, come è noto, è necessario elaborare miliardi di informazioni elementari in tempi brevissimi.

L'Amiga 2000, equipaggiata da una speciale scheda Transputer, verrà sperimentalmente impiegata in Germania nello studio di strutture molecolari complesse con lo scopo di apportare validi contributi nella ricerca chimica, biotecnologica ed alimentare.

Super camper per Amiga

In Germania è ormai completato l'allestimento di un sofisticato camper attrezzato con Amiga 2000 e PC (Commodore, naturalmente) che verrà utilizzato per gestire in tempo reale trasmissioni televisive di tipo prevalentemente sportivo.

Una delle prime fatiche cui verrà sottoposto il camper sarà la ripresa in diretta delle squadre di calcio formate dai campioni di sci (tra cui il nostro eroe Tomba) nell'ambito di manifestazioni agonistiche amichevoli a livello europeo.

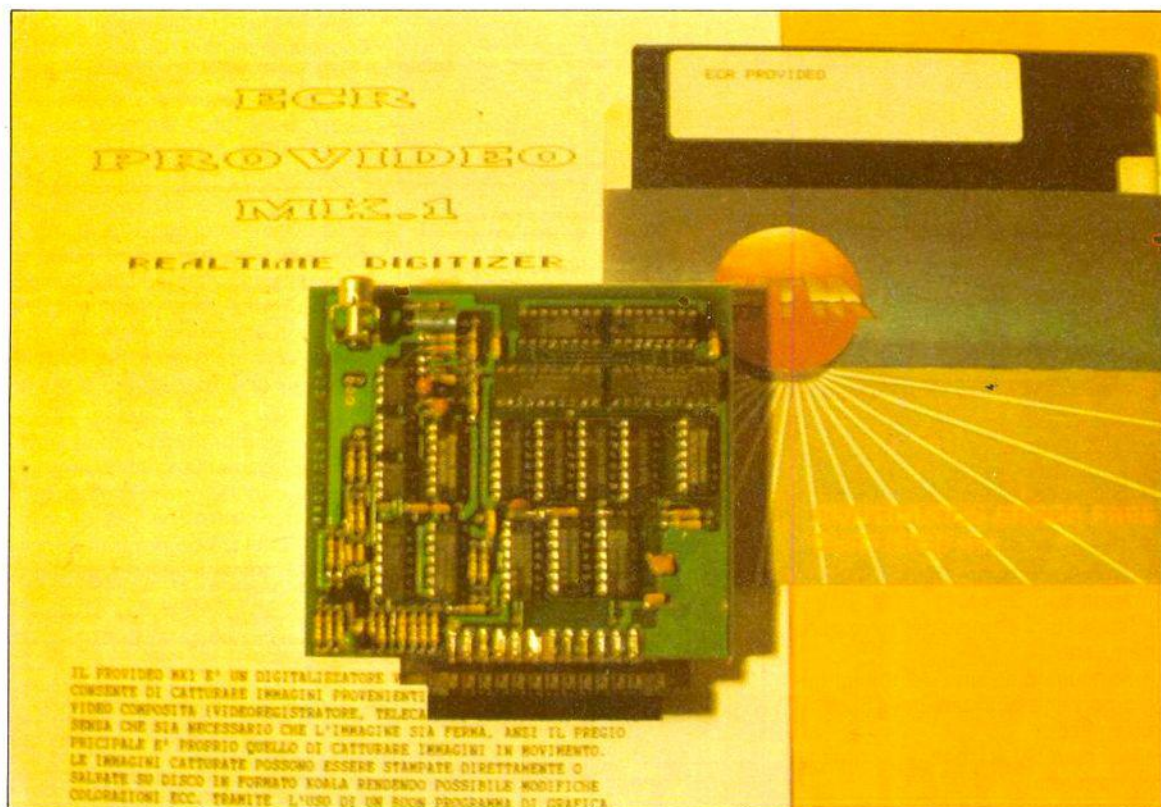
A Vaduz, infatti, si disputerà prossimamente il mini - torneo tra Italia, Germania, Svizzera ed Austria che vedrà impegnati i migliori campioni delle nevi.

Ai computer Commodore spetterà il compito, tra l'altro, di far subito apprire in sovrimpressione i dati relativi al campione che si sarà distinto in una particolare azione; potranno anche esser riproposte, durante la trasmissione in diretta, le azioni significative realizzate, a tempo di record, con animazioni computerizzate.

Amiga 2000 in italiano

Dopo la pubblicazione del manuale Amiga 500, ecco ora freschi di stampa i manuali di A-2000 che sostituiranno quelli inglesi o tedeschi delle precedenti confezioni.

Come già avvenuto per il modello A-500, coloro che intendessero entrare in possesso del volume devono inviare alla Commodore Italiana (Via F.lli Gracchi n.48 Cinisello Balsamo - Mi; tel. 02/61.83.21) la fotocopia della garanzia originale Commodore Italiana dell'A-2000.

Anche in questo caso, infatti, il "trattamento" di favore è riservato esclusivamente ai clienti entrati in possesso di A-2000 attraverso i canali ufficiali: i canali cosiddetti "paralleli" continuano ad essere penalizzati.

LA TV NEL TUO COMPUTER

Chi non è molto bravo a disegnare non può sfruttare le potenzialità dell'alta risoluzione offerte dal C/64. Ma disponendo di un digitalizzatore video in tempo reale...

di **Alessandro de Simone**

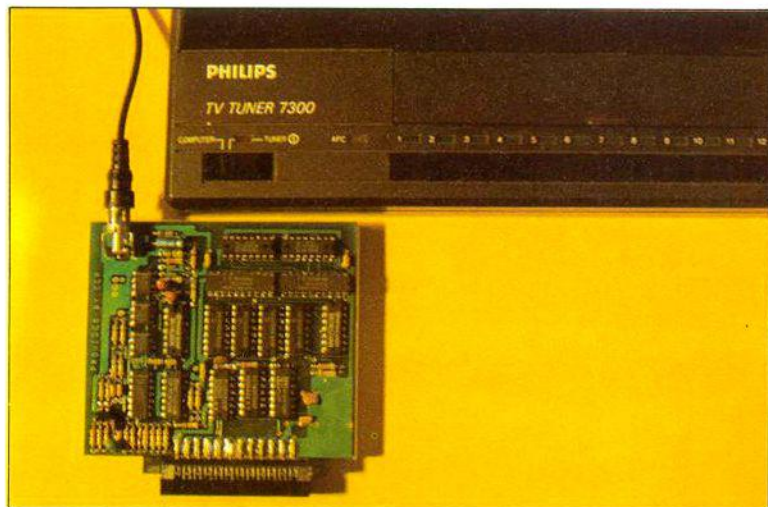
Tempo fa ebbi occasione di presentare, su queste stesse pagine, un fantastico digitalizzatore video di tipo statico (C.C.C. n.41, aprile '87).

Il "difetto" cui si accennava nell'articolo (e mi riferisco all'indispensabile staticità dell'oggetto, durante la ripresa), è stato superato da una nuova scheda proposta ancora dalla Niwa di S.San Giovanni (tel. 02/ 26.20.312) posta in vendita ad un prezzo inferiore alle 200 mila lire.

La scheda ECR Provideo MK-1, infatti, è un vero e proprio digitalizzatore dinamico che opera in tempo reale alla velocità di 90 fotogrammi al minuto oppure, se preferite, alla velocità di oltre 96000 bit al secondo, tenendo presente che ogni schermata hi-res è costituita da 8000 byte (ciascuno di 8 bit) e che i fotogrammi "catturati" sono tre ogni due secondi circa; per non parlare delle complesse elaborazioni relative ai toni dei grigi, complessivamente quattro (bianco, grigio 1, grigio 2, nero).

In pratica non si deve fare altro che collegare, mediante un comunissimo cavetto bipolare, l'uscita video del televisore, del videoregistratore, oppure del sintonizzatore TV (o di un secondo computer), alla presa sistemata sulla scheda che, dimenticavamo di dirlo, deve esser inserita nella User Port del C/64.

A tutto il resto provvede il software, fornito su dico, che permette di:

- Far scorrere le immagini digitalizzate di continuo, oppure una alla volta.

• "Bloccare" l'immagine che in quel momento compare, in modo da elaborarla con calma.
• Tarare i toni di grigio dell'immagine catturata.
• Sostituire ciascuno dei toni di grigio con uno qualsiasi dei 16 colori disponibili.
• Riversare su stampante l'immagine catturata.
• Registrare fotogrammi su disco o caricare immagini digitalizzate in precedenza.

In ogni momento, inoltre, sono disponibili alcune schermate di aiuto, in italiano.

USANDO

La prima impressione che si ha, durante il susseguirsi di immagini che riempono lo schermo, è di stupore, sia per la semplicità di uso che per la definizione delle immagini ottenute. E' ovvio che i risultati sono eccellenti durante la trasmissione di cartoni animati (che sono costituiti da fotogrammi disegnati in modo semplice e con saturazione di colore ottimale), ma anche le immagini dei normali telefilm e spot pubblicitari vengono ripresi con incredibile precisione.

Grazie alla notevole rapidità di elaborazione, sembra di assistere ad uno di quei "trucchi" di ripresa, utilizzati spesso nei Video Clip, in cui la cadenza dei fotogrammi è sostenuta in tempo reale, ma a velocità ridotta.

Il divertimento maggiore, inutile dirlo, consiste nel caricare, le immagini registrate, in ambiente Koala ed apportarvi tutte le modifiche desiderate. Su disco sono presenti anche alcuni programmi di utilità per convertire, i file-immagine, dal formato Koala in altri formati più o meno conosciuti.

Il breve programma dimostrativo, in Basic, contenuto nel disco, rappresenta, infine, un valido suggerimento per coloro che intendano far caricare, nei propri programmi, numerose immagini digitalizzate, l'una dopo l'altra.

DIGITALIZZARE E' FACILE

La tecnica detta "Digitalizzazione" consiste nel trasformare un'immagine analogica (visualizzata mediante segnali elettronici variabili con gradualità) in segnali digitali (in cui esistono solo valori unitari, cioè 1, oppure nulli, cioè 0).

Le immagini che noi siamo soliti osservare durante le trasmissioni televisive, le riproduzioni di video-nastri oppure, in diretta, mediante una telecamera collegata ad un monitor, sono di tipo analogico. Serve quindi un circuito elettronico, supportato da opportuno software, che stabilisca se un certo punto dell'immagine, dal tono ben preciso (se considerato analogico) ma indefinibile (dal punto di vista digitale) debba esser considerato bianco, grigio oppure nero.

Il compito da svolgere, pertanto, non è assolutamente semplice perchè la scheda deve scandire, rigo per rigo, l'intera immagine analogica, convertirla, in totale, in 64000 bit (caso del C/64) e stabilire, bit per bit, il colore da assegnare!

Tale difficoltà di elaborazione costringeva, fino ad un anno fa, alla ripresa di oggetti assolutamente immobili perchè il tempo impiegato dalla routine era di circa cinque secondi, durante i quali l'immagine doveva restare identica a se stessa.

Con la vecchia scheda, pertanto, era possibile digitalizzare ottime immagini mediante l'uso di una telecamera (puntata su soggetti immobili) oppure collegandola ad un videoregistratore, purchè dotato di un fermo immagine impeccabile.

La scheda di cui si parla in questo articolo, invece, consente di trasferire, senza problemi, immagini provenienti, in diretta, da normali trasmissioni TV!

PANASONIC
KX-P1081
+ COMMODORE 64
= GRAFICA HI-RES
...ED ANCHE PER
AMIGA E
COMPUTER XT-AT
COMPATIBILI

UNA STAMPANTE COME JOLLY

La stampante Panasonic KX-1081 è economica e veloce; può esser collegata sia al C/64 che all'Amiga; oppure ad un compatibile XT/AT

di **Alessandro de Simone**

Con il progresso tecnologico, le periferiche, ed in particolare le stampanti, hanno subito migliorie notevoli e, parallelamente, il loro prezzo è diminuito sempre più.

Il mercato, anzi, offre una tale varietà di modelli che l'utente finale non sa assolutamente quale scegliere e, a volte, tende ad acquistare modelli che non sono in grado di soddisfare le proprie esigenze.

Chi ha un C/64, ovviamente, vuole soprattutto sapere se una certa stampante è compatibile con il proprio computer perchè, come è noto, lo standard Commodore è piuttosto particolare e non sempre si adatta alle caratteristiche delle stampanti poste in commercio.

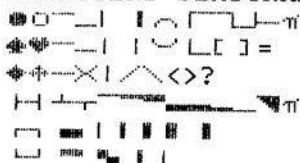
La cosiddetta compatibilità diventa un problema di difficile soluzione dal momento che non tutti i rivenditori consentono di effettuare prove ed il cliente è costretto ad acquistare a scatola chiusa con le probabili conseguenze che è facile immaginare.

Se una stampante è però venduta con l'etichetta "803 - compatibile", si può esser sufficientemente sicuri che funzionerà bene con i vari programmi, soprattutto grafici, appartenenti alla ricca collezione del C/64.

SCHEDA TECNICA

Standard Centronics
Velocità (cps= caratteri per secondo):
120 cps (modo normale)
24 cps (NLQ)
Dimensioni carta (mm.):
larghezza da 102 - 229
altezza da 127 - 363
n.copie eseguibili: 3
Aghi: 9
Matrice carattere:
9 x 9 (normale)
18 x 18 (NLQ)

```
FRASE NORMALE
FRASE ESPANSA
FRASE IN REVERSE
ECCO IL MODO GRAFICO:

CARATTERI SEMIGRAFICI:
```

```
100 rem prova stampante panasonic kx-1081"
110 rem compatibilita' codici mps-803
120 :
130 open 1,4:rem apertura can. standard
140 print#1,"frase normale"
150 print#1,chr$(14)"frase espansa"
160 print#1,chr$(18)"frase in reverse"
170 print#1,"ecco il modo grafico:"
180 print#1,chr$(8);:rem modo grafico
190 forj=1to5:for i=128 to 255
200 print#1,chr$(i);:next:print#1
210 nextj
220 print#1,chr$(15):rem ritorno norm.
230 print#1,"caratteri semigrafici:"
240 print#1,"QWERTYUIOP⌐╱─✕":rem shift
250 print#1,"ASDFGHJKL[ ]=":rem shift
260 print#1,"ZXCVBNM<>?":rem shift
270 print#1,"├┤┴┬▔▔▁▁▒▒":rem commod.
280 print#1,"┌┐ ▬ ▌ ▌▐ █ █":rem commodore
290 print#1,"└┘ ▀ ▚ ▌ ▏":rem commodore
300 print#1:close1
310 open1,4,7:rem selez. maiusc/minus.
320 cmd1:list:rem hard copy listato
330 print#1:close1
```

INTERFACCE E COMPATIBILITA'

Molto spesso abbiamo messo in guardia i nostri lettori sul controllo della reale compatibilità di un certo modello di stampante; vediamo ora dove risieda realmente il problema.

Quando un computer invia caratteri alla stampante questa deve riprodurli così come l'utente si aspetta che siano. Già in questa fase sorgono i primi problemi (per le stampanti non originali Commodore) perchè il codice Commodore è piuttosto diverso da quello ufficiale ASCII; una delle particolarità risiede nella diversità di codice dei caratteri minuscoli e maiuscoli e nei vari caratteri speciali (cursore, cancellazione, colore) che non hanno equivalente nello standard ASCII.

Il primo problema, pertanto, è quello di affidare ad un software particolare il compito di effettuare la corretta "conversione" di codici.

A questo provvede, di solito, l'interfaccia, ovvero quell'apparecchietto elettronico che si deve frapporre tra il computer e la stampante; al suo interno, infatti, non solo è presente la circuiteria necessaria per il trasferimento dei dati, ma anche una Eprom che racchiude i codici originali Commodore ed i corrispondenti ASCII "veri".

Una stampante 803-compatibile, che quindi ha al suo interno la circuiteria necessaria, forse rappresenta un vantaggio per l'utente che possiede il solo C/64; questi, infatti, deve limitarsi a collegare un cavetto tra il computer e la stampante per vederla funzionare come se fosse una Mps-803 Commodore.

Tuttavia questi tipi di stampanti possono essere utilizzati solo con il C/64; è molto meglio, invece, disporre di una stampante dotata di standard Centronics e corredata di interfaccia esterna, da collegare al C/64.

L'interfaccia Centronics è appunto fornita con la stampante di cui ci stiamo occupando ed è valida anche per trasferire dati verso altre periferiche che richiedano lo stesso formato; sarà possibile, ad esempio, inviare dati ad un plotter, se mai ve ne dovesse capitare uno, e ad altre apparecchiature che offrono le potenzialità dello standard universale Centronics.

Se, poi, si possiede anche un Amiga oppure un PC/XT/AT compatibile (standard Ms-Dos) sarà sufficiente staccare l'interfaccia Centronics-Commodore e collegare direttamente, al suo posto, il cavetto di do-

July 1988

SON	MON	DIE	MIT	DON	FRE	SAM
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

tazione: questa operazione, come intuitivo, non sarebbe possibile con stampanti che, "trasformate" fin dall'inizio in 803-compatibili, hanno al loro interno l'interfaccia non "estraibile".

Con una sola stampante, ed un'interfaccia esterna, si può disporre, insomma, di un versatile strumento di stampa in grado di essere collegato senza problemi a qualunque computer esistente (IBM, Olivetti, Atari, Apple, Amiga, C/64, C/128, C/16 e così via).

LE CARATTERISTICHE

Sulla Panasonic KX-P1081, oltre alla possibilità di selezionare il trascinamento a rullo oppure con il trattore, è interessante il selettore a tre posizioni, posto sul lato anteriore della macchina, che permette il modo di stampa "normale", il modo NLQ (Near Letter Quality: stampa simile a quella di una macchina da scrivere) ed il modo "compresso" che permette di stampare fino a 137 caratteri per riga (invece dei soliti 80) in un formato, ovviamente, ridotto.

I tre pulsanti sulla destra servono, rispettivamente, per far avanzare il rullo di un rigo alla volta, di un foglio alla volta e di interrompere il collegamento con il computer durante la fase di stampa. Questo accorgimento è utilissimo per consentire il cambio di fogli, la loro corretta sistemazione, la regolazione della pressione della testina e così via; ad una successiva pressione del tasto verde la stampa procederà esattamente dal punto in cui era stata interrrotta.

COMPATIBILITA'

La macchina in prova è stata sottoposta ai consueti test cui sottoponiamo le stampanti Commodore compatibili.

Ci riferiamo al word processor Easy Script (di cui rispetta tutti i comandi speciali di formattazione e di

LO STANDARD MPS-803

Lo standard Commodore, come è noto, è diverso da altri standard presenti sul mercato.

Quando si verifica la necessità di provare apparecchiature non originali, pertanto, è doveroso sottoporre la periferica in prova ad una serie di test che la vedano impegnata come se fosse una macchina Commodore.

In altre parole, nell'effettuare le verifiche del caso, ci chiediamo:
"Questa periferica offre gli stessi risultati che offrirebbe lo standard originale?"
Oppure, nel caso particolare:
"La stampante in prova si comporta esattamente allo stesso modo di come si comporterebbe la Mps-803, oppure necessita di particolari accorgimenti per un corretto funzionamento?"

Per rispondere onestamente a queste domande, l'unica azione che si può intraprendere è quella di utilizzare i programmi che compaiono sul libretto di istruzioni Commodore e di farli girare dopo aver collegato la stampante compatibile.

Un altro doveroso accorgimento è quello di far girare programmi grafici di una certa complessità e di confrontare i risultati ottenuti con quelli forniti dalla stampante "originale".

Inutile dire che, in entrambi i casi, la Panasonic KX-1081 ha rispettato egregiamente le regole (ovviamente ad una velocità maggiore) perfino con il programma Print Master. Questo famoso package, come forse molti sapranno, si è rivelato, infatti, un osso particolarmente duro per alcune stampanti dichiarate 803 compatibili che, per motivi misteriosi (ma inevitabilmente legati alla compatibilità) si impiantavano al momento della stampa.

Con il modello in prova, invece, tutto si è svolto regolarmente e possiamo affermare, senza ombra di dubbio, che la compatibilità vantata è reale e completa.

Inutile dire che anche la compatibilità con lo standard Centronics è più che garantita!

PANASONIC KX-P1081

una stampante universale per tutte le esigenze

PANASONIC KX-P1081 & ALDUS PAGEMAKER

Come si puo' notare da questo semplice esempio, la stampante Panasonic, dotata di interfaccia Centronix standard, e' totalmente e perfettamente compatibile con programmi professionali quali Windows e Pagemaker; package, questi, supportati da un qualsiasi computer At-compatibile.
La ripetizione di questo breve testo, scritto "sotto" Easy Script (per C/64) e trasferito via Rs-232 al computer AT, per poi essere ulteriormente elaborato, dimostra, senza ombra di dubbio, la notevole versatilita' dei prodotti citati.

PANASONIC KX-P1081 & DESKTOP PUBLISHING ALDUS PAGEMAKER

Come si puo' notare da questo semplice esempio, la stampante Panasonic, dotata di interfaccia Centronix standard, e' totalmente e perfettamente compatibile con programmi professionali quali Windows e Pagemaker; package, questi, supportati da un qualsiasi computer At-compatibile.
La ripetizione di questo breve testo, scritto "sotto" Easy Script (per C/64) e trasferito via Rs-232 al computer AT, per poi essere ulteriormente elaborato, dimostra, senza ombra di dubbio, la notevole versatilita' dei prodotti citati.

Provo effettuato con computer AT-compatibile Epson PC AX-2

PANASONIC KX-P1081 & ALDUS PAGEMAKER

Come si puo' notare da questo semplice esempio, la stampante Panasonic, dotata di interfaccia Centronix standard, e' totalmente e perfettamente compatibile con programmi professionali quali Windows e Pagemaker; package, questi, supportati da un qualsiasi computer At-compatibile.

La ripetizione di questo breve testo, scritto "sotto" Easy Script (per C/64) e trasferito via Rs-232 al computer AT, per poi essere ulteriormente elaborato, dimostra, senza ombra di dubbio, la notevole versatilita' dei prodotti citati.

scrittura); ai programmi grafici "Print Shop" e "Print Master" (che sfruttano intensivamente la pagina grafica e ne riproducono l'Hard Copy); e ad altri programmi per C/128, in Basic e l.m, che generano figure in alta risoluzione.

Inutile dire che i risultati sono da considerare ineccepibili.

La stessa stampante è stata collegata, dopo aver semplicemente staccato il cavetto dell'interfaccia, ad un computer Epson AT compatibile. Oltre ai soliti programmi è stato utilizzato anche Windows (che consente di effettuare numerose operazioni in grafica) e al Desk Top Publishing Aldus Pagemaker.

I risultati sono certamente notevoli, soprattutto tenendo conto che la stampante è a "soli" nove aghi e che il prezzo di vendita è modesto.

CONCLUSIONI

La stampante Panasonic KX-1081 è certamente una macchina valida, veloce, sicura, affidabile e, soprattutto, in linea con i prezzi che offre il mercato; se si aggiunge il fatto che il suo nome è Panasonic (da anni leader in numerosi settori dell'elettronica avanzata), ne risulta un giudizio globale decisamente favorevole.

Ad un prezzo inferiore è forse possibile trovare altri modelli, ma è doverosa una riflessione sulla reale opportunità di risparmiare qualche biglietto da diecimila.

L'interfaccia Centronics, inoltre, permette di collegare, ad un semplice C/64, una stampante di caratteristiche medio-superiori, distanti, cioè, dallo standard limitatamente hobbystico di altre apparecchiature.

Se, poi, l'utente C/64 vorrà "passare" all'Amiga oppure al pianeta Ms-Dos, si ritroverà a possedere di già una periferica pronta da collegare. .

E scusate se è poco.

Per maggiori informazioni consigliamo di rivolgersi ai ditributori regionali autorizzati:

Fanton Milano
Via Melegnano, 20
20019 Settimo Milanese
Tel. 02/32.87.312

Fanton Torino
Via Cimabue, 5
10137 Torino
Tel: 011/30.97.347

Audiomusi
Via Melato, 3
42100 Reggio Emilia
Tel: 0522/55.60.40

Tre più
Via M. Peroglio, 15
00144 Roma
Tel: 06/59.84.841

P.D.S.
Via Sicilia, 7
06085 Perugia
Tel: 075/75.10.56

T.D.P.
Via San Carlo, 14
80126 Napoli
Tel: 081/40.17.77

I.S.M.O.
Via De Magistris, 7/9
Tel: 070/65.02.90

E' ARRIVATA AMIGAZZETTA

Un superdischetto con centinaia di file è il risultato di una nuova iniziativa editoriale

di **Luigi Callegari**

Il nostro gruppo editoriale ha deciso di rispondere alle sempre crescenti richieste di software di qualità per Amiga producendo un dischetto 3.5 grazie anche alla collaborazione dei lettori.

Tempo fa, infatti, avvertimmo della partenza del progetto "Mille Icone", tuttora valido, con cui invitavamo i lettori ad inviare le loro icone di Workbench, create con IconED e colori standard.

Molti hanno inviato lavori graficamente interessanti e, di conseguenza, prontamente inseriti sul dischetto. Tra gli altri, vogliamo ringraziare e segnalare in particolare: Giancarlo Castello di Ventimiglia, Egidio Napolitano di Taranto e Paolo Manardelli di Ravenna.

Oltre alle icone abbiamo inserito nel disco, battezzato AMIGAZZETTA, molte altre cose: programmi in C ed in Assembler, completi di documentazione in italiano ed alcune fonti di caratteri da usarsi con Notepad o programmi grafici.

Il disco, per nulla protetto, è stato realizzato per essere usato su qualsiasi modello di Amiga: 500 / 1000 / 2000 con uno o con due drive e Kickstart versione 1.2. E' dotato di partenza automatica (cioè si può inserire alla richiesta del Workbench) ma è utilizzabile anche da Workbench.

Nonostante la grande quantità di cose inserite (più di trecento file!), il prezzo è assai contenuto: solo lire 15000 spese di spedizione comprese.

ICONE

Suddivise in ben sette gruppi, comprendono complessivamente oltre centocinquanta immagini di vario formato, pronte per essere agganciate ai vostri programmi da usare tramite Workbench. Nel disco sono comprese anche estese istruzioni di uso, richiamabili in ogni istante da Workbench (o CLI), con chiarimenti pratici circa l'effettivo modo di collegamento di una icona ad un programma applicativo.

Molte immagini sono veramente dei piccoli capolavori e, talvolta, dotate di doppia immagine, per indicare icona selezionata o meno (create usando IconED e poi IconMerge del disco di Workbench).

Naturalmente l'utente potrà apportare ulteriori modifiche ai simpatici disegnini proposti.

FONTI

Vi sono 12 fonti di caratteri di vari formati, pronte per essere inserite nel disco di Workbench personalizzato. Il loro formato è standard e sono utilizzabili, quindi, da Notepad, Deluxe Paint e tutti i programmi grafici e di Desktop Publishing per Amiga.

Sul disco sono comprese note d'u-

so a video, richiamabili in ogni istante.

PROGRAMMI

Le directory C e System comprendono ben 6 nuovi programmi in C od Assembly. Per ogni programma sono presenti sul dischetto delle complete note d'uso, richiamabili a video da Workbench o CLI. Questi nuovi programmi sono:

SHOW - Attivabile da Workbench e CLI, permette di visualizzare qualunque file grafico in formato IFF e HAM istantaneamente ed anche in sequenza. Si tratta di un programma sviluppato in linguaggio C da Carolyn Scheppner dello staff tecnico Commodore, appartenente al "pubblico dominio", ultima versione (supporta anche modi interlacciati!).

IconED2 - Nuova utility per la creazione di Icone, versione differenziata dell'originale Commodore. Non comprende il modo testo, ma include la possibilità di creare figure poligonali (cerchi, rettangoli...) e di inserire figure alternate per la selezione automatica (come con IconMerge). Appartenente alla produzione di pubblico dominio americana.

VERIFY - Un elegante programmino CLI che verifica la perfetta leggibilità di tutti i settori di un dischetto, con segnalazioni in italiano estese di eventuali errori. Si tratta di una produzione americana di pubblico dominio in linguaggio Assembly, tradotta e migliorata da Luigi Callegari.

ECHO - Un possibile sostituto dell'omonimo programma di CLI, scritto in linguaggio Assembly, che permette di fare generare da file batch delle scritte nella attuale finestra CLI con vari stili (grassetto, sottolineato...) e colori. Appartenente alla produzione americana di pubblico dominio.

CRYPT - Permette di codificare (rendere illeggibili od ineseguibili) e decodificare istantaneamente, da CLI, qualunque tipo di file, tramite una parola chiave. Il particolare algoritmo garantisce oltre nove miliardi di possibilità di crittografazione per ogni file. Si tratta di un programma originale, ideato e scritto da Luigi Callegari in linguaggio C.

Speriamo di aver fornito un'idea esauriente di tutto ciò che abbiamo inserito in questo nostro numero zero di AMIGAZZETTA. Se avrà il vostro apprezzamento, provvederemo in futuro a realizzarne altri, anche con la graditissima collaborazione dei lettori.

Chi volesse inviare icone, fonti di caratteri, programmi (completi di testo sorgente) potrà vedere il frutto del proprio lavoro pubblicato sui dischetti Systems per Amiga e fare notare la propria abilità a tanti altri amici lettori.

UNA DIETA SCIENTIFICAMENTE CRUDELE

Il C/64 vi aiuta a individuare la quantità di cibo necessaria per soddisfare le imposizioni di un qualunque tipo di dieta

di **Alessandro de Simone**

Quasi sempre le riviste specializzate (leggi: femminili) affrontano il problema della linea soltanto nei mesi precedenti l'estate; ciò viene fatto per consentire alle lettrici di provvedere per tempo ad occultare il lardo che altrimenti straripererebbe dai bordi dei sempre più ridotti costumi da bagno.

I mesi di marzo, aprile e maggio sono tradizionalmente dedicati a diete dai nomi più assurdi, come dieta semaforo, mediterranea, gastropatica, anti-trippa, anti-cellulite, e perfino la macabra "dieta finale".

Durante l'estate il problema sembra non più sussistere, forse perchè, ormai, ciò che è fatto è fatto ed i sacrifici primaverili cedono il passo a bestiali cene a base di sughi, intingoli e dolci vari, conditi da superalcolici degni dei bassifondi londinesi.

SCHEDA TECNICA

Software di utilità "domestica"

Hardware richiesto: C/64, ma facilmente adattabile ad altri computer Commodore

Consigliato ai principianti

Anche il programma pubblicato in queste pagine è contenuto nel disco "Directory" di questo mese.

Ci si dimentica, insomma, che il problema della dieta non dovrebbe esser considerato stagionale nè, tantomeno, che il suo fine ultimo sia (solo) quello di entrare nei jeans dell'anno scorso.

Una dieta sana, degna di questo nome, deve tener conto delle esigenze di colui che deve seguirla e non può limitarsi solo al conteggio delle calorie contenute nei cibi.

Purtroppo, però, quasi mai, sulle riviste femminili, vengono citati gli altri "costituenti" dei cibi che contribuiscono in maniera determinante al corretto equilibrio alimentare.

Nei casi, insomma, in cui si desideri considerare protidi, lipidi, il contenuto di fosforo o di ferro (tanto per citare gli elementi più comuni) nessuno sembra in grado di aiutarci; tranne, ovviamente, noi di Commodore Computer Club (e scusateci se è poco).

CHE COSA C'E' NEL CIBO

Mi colpì, (tanto, tanto tempo fa...) un libretto diffuso gratuitamente nelle scuole da parte del Ministero dell'agricoltura nel corso di una campagna per una sana ed economica alimentazione. In esso erano riportate tabelle relative alle sostanze nutritive contenute in tutti i cibi facilmente rintracciabili.

Il problema, che passai ai nostri ottimi collaboratori Mariani e Sorgato, consisteva nel computerizzare le tabelle da manipolare opportunamente in modo che l'utente finale, prima di sedersi a tavola, digitando il tipo e la quantità delle portate del pasto scelto, si rendesse meglio conto dei reali pericoli in cui stava per incorrere la propria linea.

Il risultato fu un programma, pubblicato qualche anno fa su una rivista del settore, che ora riproponiamo in una versione ampliata e migliorata.

CHE COSA FA IL PROGRAMMA

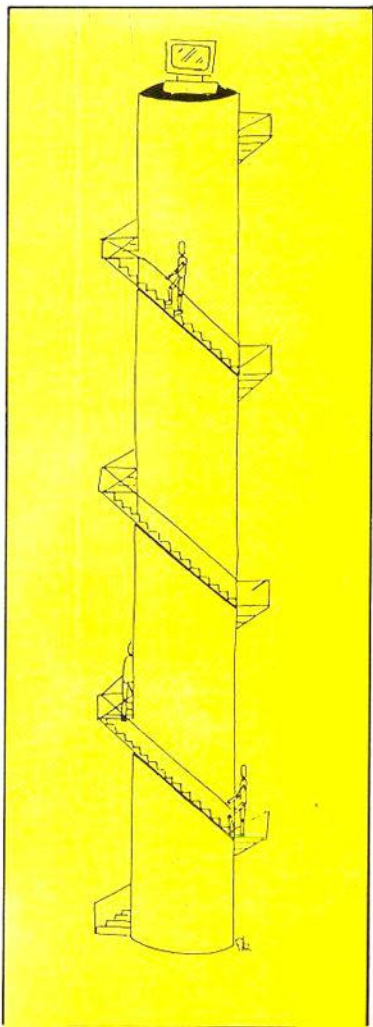
Non appena si digita il RUN vi è una piccola attesa necessaria affinchè il computer inserisca in memoria i dati relativi ai numerosi cibi. Successivamente appare il menu:

1) Lista e scelta degli alimenti
2) Totale
3) Fine
4) Lista degli alimenti scelti

E' ovvio che all'inizio si risponderà "1", in modo da esaminare i cibi a disposizione. Nel programma pubblicato questi sono 36 (vedi riga 140) e le varie elaborazioni, con minime modifiche, possono esser riportate anche su carta.

Facile è immaginare l'utilità di una tale procedura per i gestori dei ristoranti che potrebbero, in tal modo, fornire un utile servizio agli avventori con problemi di cellulite.

Tornando al programma, questo, effettuata la scelta del cibo e determinata la sua quantità, visualizza la percentuale di parte non edibile (quella che cioè non viene mangiata, come ossa, bucce, eccetera) e le seguenti... cose:

Protidi, Lipidi, Glicidi (espressi in

grammi); Calorie (unità); Calcio e Fosforo (espressi in milligrammi); Ferro (microgrammi; Vitamina A (Unità); Vitamine B1, B2, PP (microgrammi); Vitamina C (milligrammi).

C'è insomma quanto basta per effettuare una scelta oculata del pasto (o digiuno) quotidiano.

E' ovvio che è possibile sommare i dati relativi a più cibi (primo, secondo, contorno, eccetera) con un massimo di dieci portate (variabile PP); con l'opzione 2 si visualizza il totale delle sostanze nutritive relative ai cibi indicati in fase di input. Questi possono essere controllati con l'opzione 4.

Si tenga presente che è necessario "rimanere" nell'opzione "Lista e scelta alimenti" finchè non si è completato il menu; una volta usciti dall'opzione, infatti, sarà possibile solo esaminare il totale e la lista degli alimenti selezionati. La successiva selezione dell'opzione "1" annulla le scelte precedenti e reinizializza il programma.

COME AMPLIARE IL MENU

Avendo a diposizione i dati relativi ad altri cibi sarà semplicissimo aggiungerli nel programma.

Tutto ciò che dovete fare è modificare il valore 36 presente in riga 140 che segnalerà, al computer, la necessità di fare spazio per ospitare altri dati.

Quindi dovete aggiungere altre righe Data, dopo la riga 1470, tenendo conto che il primo DATA deve rappresentare il nome del cibo (in modo da poterlo, in seguito, visualizzare) mentre i successivi 13 dati (numerici) rappresenteranno i valori relativi a percentuale non edibile, protidi, lipidi glicidi (e così via), da digitare nell'ordine indicato dalle righe 550-670, per un totale, appunto, di 13 dati.

Nel caso in cui i valori siano tali da non essere ospitati in una sola riga Basic è possibile "spezzarli" su due righe, come si può constatare, ad esempio, per le alici sott'olio (righe 1200-1210).

I più bravi possono inserire una routine di hard-copy per bassa risoluzione in modo da riportare su stampante le varie schermate elaborate.

Consigliamo la routine l.m. (rilocabile) pubblicata sul N. 51 di C.C.C. nella rubrica "Enciclopedia di routine in linguaggio macchina".

DIGITANDO E PROVANDO

Nonostante siano assenti comandi pericolosi (Poke, Sys) il listato è da digitare con la massima attenzione perchè un'errata trascrizione potrebbe portare allo scambio dei dati tra loro.

Consigliamo, quindi, di farsi aiutare da un amico nella digitazione e, soprattutto, di non confondere il carattere di virgola (,) con il punto (.). Nelle righe Data, infatti, la virgola indica la separazione tra due dati numerici attigui, mentre il punto, come dovrebbe esser noto, stabilisce la parte decimale di un valore numerico. In ogni caso, dopo il nome di un cibo (racchiuso tra apici) devono esser presenti 13 valori numerici.

Se, dato il Run, dovesse comparire un Syntax error relativo ad una riga contenente un Data, controllate sia quella riga sia, soprattutto, quelle immediatamente PRECEDENTI: il motivo della segnalazione, infatti, è dovuta ad una scorretta trascrizione di virgole e punti che si ripercuote sull'assegnazione della variabile stringa successiva.

Per sincerarvene, infatti, digitate la riga 1020 come segue:

```
1020 data "pasta all'uovo", 14.7, 2.74, 67.9, 363.594, 22, 199, 2100, 200, 200, 110
```

...in cui, cioè, invece di 14,7 è presente 14.7.

L'errore segnalato in questo caso è addirittura relativo a due righe dopo (Syntax error in 1040) nonostante questa riga sia priva di errori.

```
ALIMENTO: MERLUZZO     140 GR.

PARTE NON EDIBILE: 24 %

PROTIDI:              19.474      GR
LIPIDI:               3.71        GR
GLICIDI:              1.064       GR
CALORIE:              117.9668
CALCIO:               35          MG
FOSFORO:              271.6       MG
FERRO:                980         MCG
VITAMINA A            0           UI
         B1           70          MCG
         B2           112         MCG
         PP           3080        MCG
         C            0           MG


ALIMENTO: BANANA      100 GR.

PARTE NON EDIBILE: 30 %

PROTIDI:              1.18        GR
LIPIDI:               .26         GR
GLICIDI:              20.53       GR
CALORIE:              91.377
CALCIO:               7           MG
FOSFORO:              28          MG
FERRO:                800         MCG
VITAMINA A            450         UI
         B1           60          MCG
         B2           40          MCG
         PP           600         MCG
         C            16          MG


ALIMENTO: PASTA ALIMENTARE     85 GR.

PARTE NON EDIBILE: 0 %

PROTIDI:              9.605       GR
LIPIDI:               1.0115      GR
GLICIDI:              62.815      GR
CALORIE:              306.12665
CALCIO:               8.5         MG
FOSFORO:              122.4       MG
FERRO:                1360        MCG
VITAMINA A            0           UI
         B1           110.5       MCG
         B2           85          MCG
         PP           935         MCG
         C            0           MG

TOTALE

PROTIDI:              30.259      GR
LIPIDI:               4.9815      GR
GLUCIDI:              84.409      GR
CALORIE:              515.47045
CALCIO:               50.5        MG
FOSFORO:              422         MG
FERRO:                3140        MCG
VITAMINA A            450         UI
         B1           240.5       MCG
         B2           237         MCG
         PP           4615        MCG
         C            16          MG

ALIMENTI SCELTI:

MERLUZZO                 140  GR
BANANA                   100  GR
PASTA ALIMENTARE         85   GR
```

```
100 REM **     UNA DIETA SCIENTIFICA      **
110 REM ** BY G. MARIANI & F. SORGATO    **
120 REM ** PER COMMODORE COMPUTER CLUB **
130 :
140 PRINT"":N=36:REM N. ALIMENTI DA MODIFICARE NEL CASO SI AGGIUNGANO CIBI
150 PRINT"ATTENDERE PREGO...":GOSUB 890:DIM Z(11),P$(11),P(11)
160 PRINT"MENU:"
170 PRINT"1) LISTA E SCELTA ALIMENTI"
180 PRINT"2) TOTALE"
190 PRINT"3) FINE"
200 PRINT"4) LISTA ALIMENTI SCELTI"
210 PRINT"SCEGLI (1-4)..."
220 GETA$:IFA$=""THEN220
230 IFA$<"1"ORA$>"4"THEN220
240 IF A$="3" THEN PRINT"CIAO, GRASSONE!!":CLR:END
250 ON VAL(A$) GOTO 300,710,240,270
260 GOTO 160
270 PRINT"ALIMENTI SCELTI:"
280 FORK=0TOPP-1:PRINT" "P$(K):PRINTTAB(23)""P(K)" GR":NEXT
290 PRINT"PREMI UN TASTO":POKE198,0:WAIT198,1:GOTO160
300 FORK=0TO11:Z(K)=0:P$(K)="":P(K)=0:NEXT:PP=0
310 IFPP>=10THENPRINT"NON + DI 10 ALIMENTI!!":GOTO160
320 PRINT"(PREMI SPACE PER PARTIRE E FERMARTI)"
330 PRINT"(PREMI Q PER FINIRE)"
340 GET A$:IF A$<>" " THEN 340
350 PRINT" 0 ) RITORNA AL MENU PRINCIPALE"
360 FOR K=1 TO N:PRINT K;") ";N$(K)
370 GET A$:IF A$=" " THEN 400
380 IF A$="Q" THEN 460
390 GOTO 420
400 GET A$:IF A$="" THEN 400
410 IF A$="Q" THEN 460
420 NEXT
430 PRINT"LA VUOI RIVEDERE? (S/N)"
440 GETA$:IFA$<>"S"ANDA$<>"N" THEN440
450 IF A$="S" THEN 310
460 PRINT"SCEGLI L'ALIMENTO CHE TI INTETRESSA..."
470 PRINT"QUALE ( 0 -"N")";:INPUT Q:IF Q<0 OR Q>N THEN 470
480 IF Q=0 THEN 160
490 INPUT"QUANTI GRAMMI";G
500 P$(PP)=N$(Q):P(PP)=G:PP=PP+1
510 Z(0)=Z(0)+PR(Q)*G:Z(1)=Z(1)+LI(Q)*G:Z(2)=Z(2)+GL(Q)*G:Z(3)=Z(3)+CA(Q)*G
520 Z(4)=Z(4)+CL(Q)*G:Z(5)=Z(5)+FO(Q)*G:Z(6)=Z(6)+FE(Q)*G:Z(7)=Z(7)+A(Q)*G
530 Z(8)=Z(8)+B1(Q)*G:Z(9)=Z(9)+B2(Q)*G:Z(10)=Z(10)+PP(Q)*G:Z(11)=Z(11)+C(Q)*G
540 PRINT"ALIMENTO: "N$(Q);"    "G"GR."
550 PRINT"PARTE NON EDIBILE:"PE(Q)"%"
560 PRINT"PROTIDI:          "PR(Q)*G
570 PRINT"LIPIDI:           "LI(Q)*G
580 PRINT"GLICIDI:          "GL(Q)*G
590 PRINT"CALORIE:          "CA(Q)*G
600 PRINT"CALCIO:           "CL(Q)*G
610 PRINT"FOSFORO:          "FO(Q)*G
620 PRINT"FERRO:            "FE(Q)*G
630 PRINT"VITAMINA A        "A(Q)*G
640 PRINT"         B1       "B1(Q)*G
650 PRINT"         B2       "B2(Q)*G
660 PRINT"         PP       "PP(Q)*G
670 PRINT"         C        "C(Q)*G
680 PRINT""TAB(27)"GRGRGRMGMGMCGUIMCGMCG";
690 PRINT"MCGMG"
700 PRINT"PREMI UN TASTO":POKE198,0:WAIT198,1:GOTO310
710 PRINT"TOTALE"
720 PRINT"PROTIDI:          "Z(0)
730 PRINT"LIPIDI:           "Z(1)
740 PRINT"GLUCIDI:          "Z(2)
750 PRINT"CALORIE:          "Z(3)
760 PRINT"CALCIO:           "Z(4)
770 PRINT"FOSFORO:          "Z(5)
780 PRINT"FERRO:            "Z(6)
790 PRINT"VITAMINA A        "Z(7)
```

Il manuale del Commodore 16

La traduzione in lingua italiana della guida all'utente per il C/16 non riporta l'indice alfabetico degli argomenti trattati nel volume.

Ne consegue, spesso, una ricerca difficoltosa dal momento che le varie istruzioni, comandi e funzioni non si succedono secondo un ordine facilmente individuabile dall'utente.

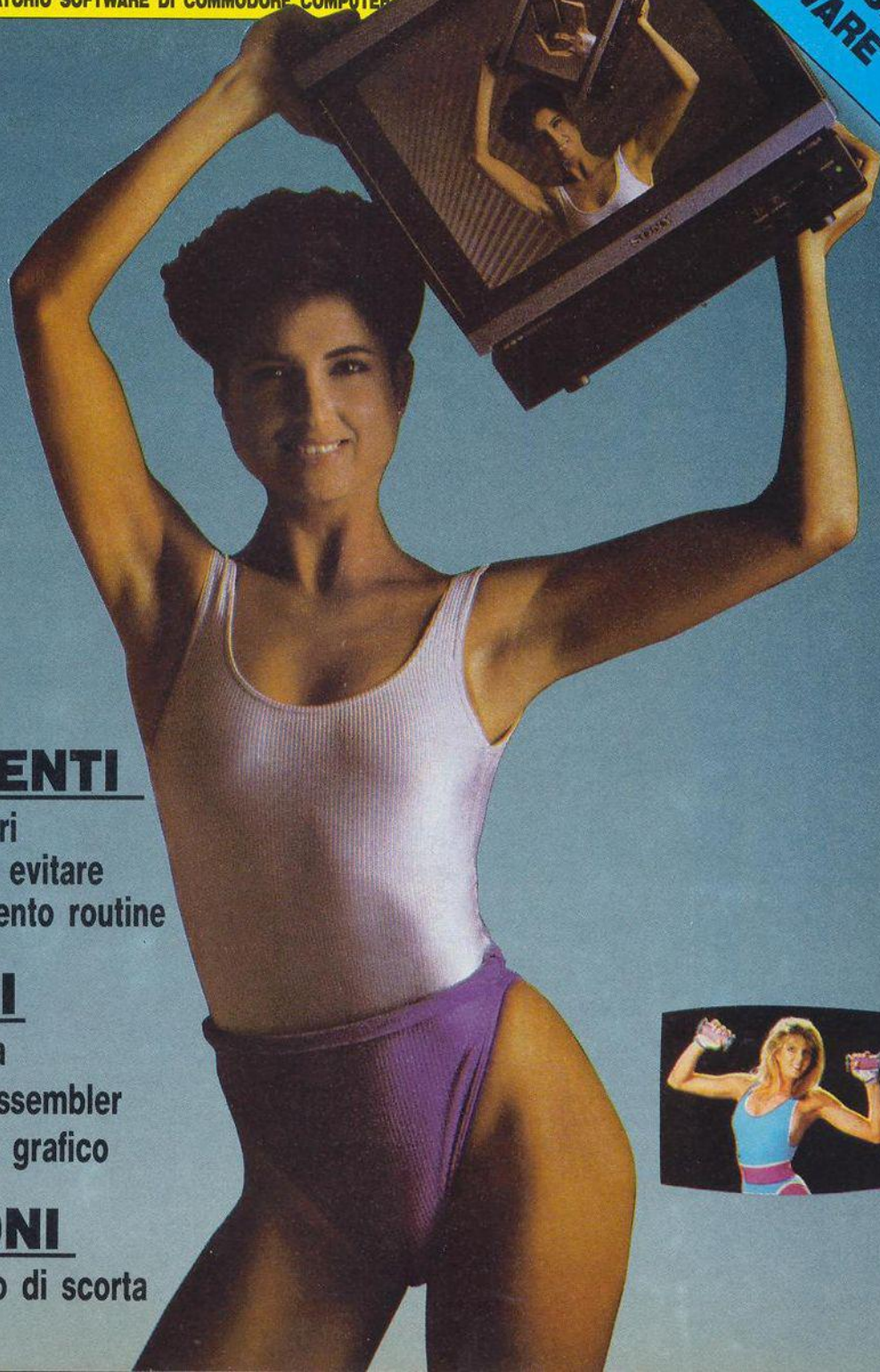
Certi di fare cosa gradita ai possessori del piccolo computer, riportiamo, qui di seguito, l'elenco alfabetico completo dei numerosi argomenti e delle pagine relative in cui sono pubblicati.

L'indice riportato all'inizio del manuale allegato al computer, invece, è sufficiente per individuare i capitoli relativi agli argomenti trattati.

```
800 PRINT"           B1    "Z(8)
810 PRINT"           B2    "Z(9)
820 PRINT"           PP    "Z(10)
830 PRINT"           C     "Z(11)
840 PRINT"■■■■■"TAB(27)"GR■■■GR■■■GR■■■■MG■■■MG■■■MCG■■■■UI■■■MCG■■■■MCG";
850 PRINT"■■■■MCG■■■■MG"
860 PRINT"■■PREMI UN TASTO"
870 POKE198,0:WAIT198,1:GOTO160
880 END
890 DIM N$(N),PE(N),PR(N),LI(N),GL(N),CA(N),CL(N),FO(N),FE(N),A(N),B1(N)
900 DIM B2(N),PP(N),C(N)
910 FOR K=1 TO N:READ N$(K),PE(K),PR(K),LI(K),GL(K),CA(K),CL(K),FO(K)
920 READ FE(K),A(K),B1(K),B2(K),PP(K),C(K)
930 NEXT
940 FOR K=1 TO N
950 PR(K)=PR(K)/100:LI(K)=LI(K)/100:GL(K)=GL(K)/100:CA(K)=CA(K)/100
960 CL(K)=CL(K)/100:FO(K)=FO(K)/100:FE(K)=FE(K)/100:A(K)=A(K)/100
970 B1(K)=B1(K)/100:B2(K)=B2(K)/100:PP(K)=PP(K)/100:C(K)=C(K)/100
980 NEXT:RETURN
990 REM  ** PRIMI PIATTI **
1000 DATA "PASTA ALIMENTARE",0,11.3,1.19,73.9,360.149,10,144,1600,0,130,100
1010 DATA 1100,0
1020 DATA "PASTA ALL'UOVO",14,7,2.74,67.9,363.594,22,199,2100,200,200,110
1030 DATA 2300,0
1040 DATA "SEMOLINO",0,12.15,0.75,76.1,368.65,16,112,1000,0,130,40,1100,0
1050 DATA "PANE DI FRUMENTO",0,8.19,0.5,62.14,292.903,14,110,1400,0,56,60,800
1060 DATA 0
1070 DATA "BISCOTTI",0,8.50,0.98,75.1,351.678,20,96,1100,0,60,50,1100,0
1080 REM  ** CARNE **
1090 DATA "CONIGLIO",20,21.5,1.5,0,101.8,17,180,1300,0,80,60,7600,0
1100 DATA "MAIALE",26,19.91,6.81,1.1,148.112,8,156,1700,0,490,140,3200,0
1110 DATA "POLLO",28,16.77,13.62,1.34,198.193,12,200,1500,400,100,160,8100,0
1120 DATA "VITELLO",22,19.86,0.82,0,88.888,11,213,2400,20,140,260,6500,0
1130 DATA "CAVALLO",36,21.71,2.55,0.46,114.102,10,200,2400,0,70,100,4200,0
1140 DATA "AGNELLO",34,20.06,2.24,0,102.630,9,191,1900,0,140,190,4700,0
1150 REM  ** PESCE **
1160 DATA "MERLUZZO",24,13.91,2.65,0.76,84.262,25,194,700,0,50,80,2200,0
1170 DATA "SGOMBRO",20,16,11.08,0.75,169.503,38,264,1200,100,80,210,6500,0
1180 DATA "TONNO",10,20.5,4.17,0.56,124.293,38,264,1200,100,80,210,6500,0
1190 DATA "CALAMARO",35,12.6,1.74,0.64,70.118,144,189,17.4,250,70,160,1400,0
1200 DATA "ALICI SOTT'OLIO",0,25.9,11.27,0.19,209.526,44,351,1300,110,60,200
1210 DATA 2600,0
1220 REM  ** FRUTTA **
1230 DATA "ALBICOCCA",12,0.49,0,8.07,35.096,15,23,500,2500,40,60,700,6
1240 DATA "ARANCIO",28,0.75,0,7.32,33.087,34,23,400,170,80,30,200,50
1250 DATA "BANANA",30,1.18,0.26,20.53,91.377,7,28,800,450,60,40,600,16
1260 DATA "PERA",18,0.3,0,12.04,50.594,13,16,300,20,20,40,100,4
1270 DATA "UVA",10,0.34,0,17.72,74.046,17,21,600,80,60,40,200,4
1280 REM  ** FORMAGGI **
1290 DATA "FONTINA",8,24.46,24.88,0.78,329.892,870,561,300,1400,10,450,100,0
1300 DATA "PROVOLONE",6,26.3,28.85,0,370.365,881,567,500,180,20,830,200,0
1310 DATA "GORGONZOLA",12,19.44,31.2,0,363.624,612,356,300,1400,10,450,100,0
1320 DATA "GROVIERA",8,31.52,31.96,0,420.068,1.123,685,500,1400,10,450,100,0
1330 DATA "PARMIGIANO",12,32.81,29.68,0,404.609,1192,695,800,950,20,610,200,0
1340 REM ** ALIMENTI VARI E BEVANDE **
1350 DATA "NOCI",65,19.25,67.78,2.87,707.49,83,380,2100,30,480,130,1200,3
1360 DATA "PROSCIUTTO COTTO",2,21.18,36.42,0.75,421.335,10,166,2900,0,540
1370 DATA 210,4200,0
1380 DATA "PROSCIUTTO CRUDO",2,18.87,46.90,0,504.157,10,136,2100,0,520,190
1390 DATA 3400,0
1400 DATA "SALAME",5,36.85,34.85,0,468.22,0,270,3700,0,230,200,2900,0
1410 DATA "UOVO DI GALLINA",12,13.06,11.13,1.06,159.175,50,210,2500,1000,100
1420 DATA 300,100,0
1430 DATA "BURRO",0,0.81,83.45,1.05,767.021,15,16,0,3200,0,0,0,0
1440 DATA "OLIO DI OLIVA",0,0,99,0,900.9,0,0,0,0,0,0,0,0
1450 DATA "LATTE DI MUCCA",0,3.5,3.5,4.63,65.183,119,93,100,140,40,180,100,1
1460 DATA "BIRRA",0,.43,0,2.70,12.833,6,0,100,0,10,30,300,0
1470 DATA "VINO",0,0,0,1.5,6.15,0,0,0,0,0,0,0,0
1480 REM INSERIRE QUI DI SEGUITO EVENTUALI ALTRI DATI
1490 REM FINE
```

CAMPUS

LABORATORIO SOFTWARE DI COMMODORE COMPUTER

LA TUA UNIVERSITA' DEL SOFTWARE

ESORDIENTI

- Tutti gli errori che vorreste evitare
- Una, dieci cento routine

ESPERTI

- Alla scoperta del Macro Assembler
- Un detective grafico

CAMPIONI

- Uno schermo di scorta

TUTTI GLI ERRORI CHE VORRESTE EVITARE

E' possibile individuare i casi particolari in cui il computer si blocca ed emette quelle stramaledette segnalazioni di errore?

di **Alessandro de Simone**

Uno dei principali motivi di avvilimento deriva, sicuramente, dalla persistente segnalazione di errore, che blocca programmi faticosamente digitati.

A volte è semplice individuare l'errore cui il computer si riferisce; altre volte, al contrario, la ricerca diventa piuttosto ardua.

Con questo articolo iniziamo la descrizione dei principali errori, partendo dalla numerazione dello standard Commodore; l'argomento è infatti troppo lungo per essere affrontato in poche pagine.

Il metodo, ovviamente, sarà il più semplice possibile: commettere gli errori a bella posta, in modo da capire il modo di evitarli in futuro.

Una situazione particolare

Prima di iniziare la trattazione sistematica, ricorderemo che il comando PRINT, ripetuto abbondantemente nei programmi di queste pagine, non può essere abbreviato con il punto di domanda (?) perchè verrà spesso riferito ad un file precedentemente aperto. Pertanto PRINT va digitato così come lo vedete pubblicato, senza ricorrere ad abbreviazioni; in caso contrario potreste vedere apparire costantemente un Syntax Error di cui non risuscireste a darvi ragione.

Non tutti i nostri lettori posseggono il drive o la stampante e la necessità di aprire un file (comando OPEN) per esaminare alcuni tipi

Un errore, spesso, è piuttosto difficile da individuare

E' bene studiare con attenzione i casi più semplici in modo da riconoscere subito situazioni anomale

di errore ci avrebbe costretto ad usare il registratore a cassette; in questo caso, però, lo schermo viene cancellato ed è necessario parecchio tempo per completare una qualsiasi procedura inviata al datassette.

Poichè lo schermo è "visto" dal computer alla stessa stregua di una qualsiasi altra periferica (verso cui inviare o da cui ricevere dati) ci serviremo del video per le nostre operazioni sui file dal momento che ognuno di noi ha a disposizione un monitor o un comune televisore (almeno ce lo auguriamo...).

In parole più semplici, il comando...

Print "Pippo"

...ha lo stesso effetto del gruppo di comandi...

Open 1,3
Print#1, "Pippo"
Close 1

Nonostante ciò possa sembrare complicato, se non addirittura inutile, vi chiediamo di digitare il primo programma che ha lo scopo di esaminare il messaggio "Too many files" dovuto, appunto, al primo errore dello standard Commodore.

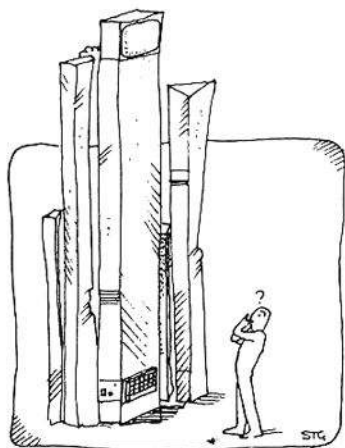
```
100 REM STUDIO DELL'ERRORE N.1
110 REM TOO MANY FILES
120 :
130 REM IMPOSSIBILE TENERE APERTI
140 REM PIU' DI 10 FILE ALLA VOLTA
150 :
160 FOR I=1 TO 10: OPEN I,3
170 REM NON SI PUO' ABBREVIARE PRINT
180 REM CON IL PUNTO DI DOMANDA (?)
190 PRINT#I,"FILE N."I:REM MESSAGGIO
200 NEXT:REM ORA SONO APERTI 10 FILE
210 OPEN 11,3:REM APERTURA 11MO FILE
220 REM BLOCCO DEL COMPUTER IN 210
```

In questo listato si suppone di aver già aperto in precedenza dieci file; naturalmente questa situazione è insolita poichè difficilmente capiterà di aprire più di un paio di file alla volta.

Quando si stampa, ad esempio, è sufficiente aprire un solo file; può capitare, tuttavia, di aprire un file in lettura sul disco e di voler riversare il suo contenuto su carta (dopo aver aperto, quindi, un secondo file verso la stampante). Nel caso in cui, contemporaneamente, si voglia elaborare il contenuto e riversarlo nuovamente su disco, oltre che su carta, il numero di file sale a tre.

Operando con i file relativi, il numero di file realmente aperti sale considerevolmente anche se, in apparenza, risulta minore.

Facendo girare il programma, dunque, viene emessa una segnalazione di errore nel momento in cui si desidera aprire l'undicesimo file; si noti bene che la riga sulla quale "cade" il computer non contiene un

```
100 REM STUDIO DEGLI ERRORI N.2/3
110 REM FILE OPEN / NOT OPEN
120 :
130 OPEN 1,3:REM FILE OPEN ERROR
140 INPUT "PRIMO ADDENDO";X
150 INPUT "SECON.ADDENDO";Y
160 IF X+Y=0 THEN 100
170 PRINT#1,"SOMMA="X+Y: GOTO130
180 :
190 REM PER EVITARE ERRORI CHIUDERE
200 REM SEMPRE UN FILE PRIMA DI
210 REM APRIRLO. ESEMPIO (RIGA 130)
220 REM CLOSE 1: OPEN 1,3
```

errore di sintassi o di altro tipo: indica solo un "Too many file" che sta a ricordare, appunto, che il numero massimo consentito di file è stato superato.

Per ovviare ad errori di questo tipo, ricordatevi sempre di chiudere un file non appena non serve più. Se l'inconveniente dovesse verificarsi ancora, modificate il vostro programma in modo da aprire un file, compiere le varie operazioni che ne hanno richiesto l'apertura e quindi chiuderlo PRIMA di aprirne un altro. In questo modo è probabile che il tempo necessario per un'elaborazione aumenti ma, purtroppo, spesso non si può ovviare all'inconveniente.

Gli errori 2 e 3

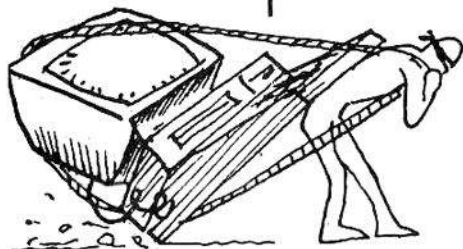
Un caso molto comune, lavorando con i file, è quello relativo alla segnalazione di file non aperto ("File not open error") e di file già aperto ("File open error"). Ciò significa che un file, una volta aperto, non può essere aperto una seconda volta (a meno che non si provveda, prima, a chiuderlo); nè è possibile, d'altra parte, scrivere qualcosa su un file non aperto: è come se voleste scrivere... in aria, senza aver preso un foglio di carta!

Se, quindi, vi capita un errore di questo genere è probabile (anzi, è sicuro) che non abbiate digitato correttamente il listato oppure che avete dimenticato di inserire una riga Basic in cui comunicare al computer l'ordine di aprire un file

Se fate partire con il solito Run il secondo programma (privo, però, della riga 130 in cui compare "Open 1,3") il computer si bloccherà subito sul primo Print dal momento che questo comando richiede una

Spesso la causa degli errori segnalati è da imputare al valore delle variabili adoperate

```
100 REM STUDIO DEGLI ERRORI 6 E 7
110 REM NOT OUTPUT / INPUT FILE
120 REM BATTERE (RUN 150 E RUN 190)
130 :
140 OPEN 1,0:REM FILE VERSO TASTIERA
150 PRINT#1,123
160 REM IMPOSSIBILE INVIARE DATI
170 REM VERSO LA TASTIERA (DEVICE:0)
180 :
190 OPEN 1,4:REM DATI DA STAMPANTE
200 INPUT#1,X
210 REM IMPOSSIBILE RICEVERE DATI
220 REM DALLA STAMPANTE (DEVICE:4)
230 REM (PREMERE RUN/STOP E RESTORE)
```

```
100 REM STUDIO DELL'ERRORE N.10
110 REM NEXT WITHOUT FOR
120 :
130 FOR I=1 TO 5: PRINT I;
140 INPUT "SCRIVI";X$
150 IF X$="FINE" THEN NEXT
160 PRINT TAB(10)"FRASE:" X$
170 NEXT: REM FINE DEL CICLO
180 PRINT CHR$(18)"FINE PROGR."
190 :
200 REM CORREGGERE LA LINEA 150:
210 IF X$="FINE" THEN 170
```

Errori molto comuni sono da attribuire all'uso troppo disinvolto dei cicli "For...Next", Read...Data" e "Gosub... Return

preventiva apertura del file verso il video. Dopo aver digitato la riga contenente Open, vedrete che il programma eseguirà la prima elaborazione, bloccandosi, però, a causa della nuova richiesta di apertura dello stesso file.

Morale: ricordatevi di non indirizzare mai un Goto o un Gosub verso un comando Open, oppure provvedete sempre a chiudere lo stesso file prima di aprirlo. Mentre, infatti, è "vietato" aprire due volte lo stesso file, è consentito chiudere più volte un file, anche se è già stato chiuso (oppure non esiste affatto).

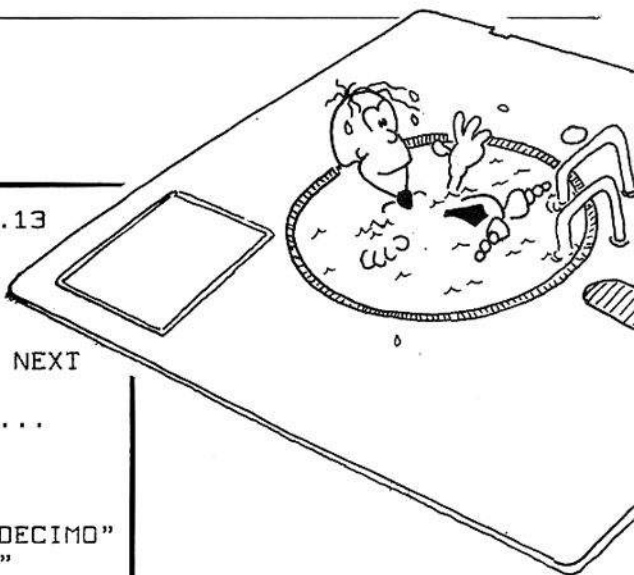
Gli errori 6 e 7

Poca attenzione meritano i segnali di errore 6 ("Not output file") e 7 ("Not input file") che vengono segnalati tentando di inviare una stringa, un numero o un "qualcosa" verso una periferica abilitata solo per inviare dati oppure, al contrario, tentar di ricevere dati da una periferica costruita solo per inviare dati.

La tastiera, che nei piccoli Commodore ha numero di device 0, è una periferica che, come è intuitivo, è in grado solo di inviare dati al computer e non di riceverne. L'errore che ne deriva, quindi, è pienamente giustificato. Facendo partire lo stesso programma con Run 200, invece, si tenta di ricevere dati dalla stampante (numero di device 4) che, come è noto, è solo in grado di ricevere dati e stamparli su carta.

Se pertanto, in altri vostri programmi, compare un errore del genere, controllate il valore del device; se questo è individuato da una variabile (come nel caso: Open X,Y) controllate la variabile stessa e verificate la

```
100 REM STUDIO DELL'ERRORE N.12
110 REM RETURN WITHOUT GOSUB
120 :
130 INPUT "SCRIVI IL TUO NOME";X$
140 GOSUB 170
150 :
160 REM INIZIO SUBROUTINE STAMPA
169 REM END
170 FOR I=1 TO 20: PRINT TAB(I) X$
180 NEXT: RETURN
190 :
200 REM OCCORRE INSERIRE RIGA DI FINE
210 REM PRIMA DELLE VARIE SUBROUTINE.
220 REM PER ESEMPIO: 169 END
```

```
100 REM ESEMPI DI VARI ERRORI N.13
110 REM OUT OF DATA
120 :
130 FOR I=1 TO 12: READ A$
140 :
150 PRINT "DATO N."I;TAB(12)A$: NEXT
160 :
170 REM SEMBRA CHE SIANO 12, MA...
180 :
190 DATA PRIMO, SECONDO TERZO
200 DATA QUARTO, QUINTO, SESTO,
210 DATA SETTIMO,,OTTAVO,"NONO,DECIMO"
220 DATA UNDICESIMO "DODICESIMO"
230 :
240 REM INSERIRE VIRGOLA (RIGA 190)
250 REM TOGLIERE APICI E VIRGOLA(210)
260 REM INSERIRE VIRGOLA (RIGA 220)
```

corrispondenza con la periferica in oggetto.

For...Next

Un errore molto comune, che spesso sembra casuale, è quello dovuto all'errata gestione del ciclo For...Next.

A volte, infatti, è sufficiente un If...Then di troppo (o di meno) per far saltare l'elaborazione.

Digitate, infatti, il programma in oggetto e fatelo partire con il solito Run; verrà chiesto di digitare un gruppo di caratteri ("Frase?"); se battete una qualsiasi parola noterete che il programma viene portato a termine senza errori; questo accade facendolo partire nuovamente, e digitando "Fine" in qualsiasi posizione tranne che alla quinta domanda; viene segnalato un errore, invece, se alla quinta domanda rispondete, appunto "Fine". In questo caso, infatti, viene attivato il ciclo If...Then ma stavolta la variabile "I" ha già raggiunto il valore 5, il Next è stato già eseguito per l'ultima volta, il ciclo è già chiuso ma... viene richiesto un nuovo Next, che genera l'errore.

Quasi sempre questa segnalazione deriva dall'inserimento di un comando Next dopo un Then. Per evitare inconvenienti che, come abbiamo visto, sono spesso subdolamente nascosti, è indispensabile non inserire mai un Next dopo un Then; meglio seguire il suggerimento dell'ultima riga del listato di esempio.

Gosub e Return

Un errore relativamente semplice da individuare è quello relativo ad un Return incontrato senza aver impartito il corrispondente Gosub.

Quasi sempre (a parte una banale dimenticanza) tale errore viene segnalato perchè ci si dimentica di apporre un comando End prima del gruppo di righe riservato alle subroutine.

I programmatori, di solito, allocano le subroutine del Basic "dopo" il programma principale ma dimenticano, molto spesso, che quando questo termina, in effetti... continua per la sua strada, incontrando le subroutine.

Il programma che esamina il caso specifico dimostra in modo semplice il caso in cui si può manifestare in errore del genere. E' sufficiente, però, seguire il semplice suggerimento indicato nell'ultima riga per sistemare tutta la faccenda.

Out of Data

Un errore che manda in bestia i principianti che intendono passare al linguaggio macchina è proprio "Out of Data", che vie-

La gestione dei file richiede un'esperienza notevole che, tuttavia, è semplice da raggiungere

```
100 REM ALCUNI ESEMPI DI ERRORE 14
110 REM ILLEGAL QUANTITY ERROR
120 :
130 REM ILLEGALI PERCHE' NEGATIVI
140 X=-12:POKE X,12     : REM INDIRIZ.
150 X=-4:  PRINT TAB(X): REM ARGOMEN.
160 X=-4:  PRINT SQR(X): REM ARGOMEN.
170 X=-45:PRINT LOG(X): REM ARGOMEN.
180 X=-23:POKE 70,X    : REM ARGOMEN.
190 :
200 REM ILLEGALI PERCHE' ECCESSIVI
210 X=256:  PRINT TAB(X):REM ARGOMEN.
220 X=32768:   DIM A(X): REM ARGOMEN.
230 X=256:  POKE 1024,X: REM ARGOMEN.
240 PRINT PEEK(123456): REM INDIRIZ.
250 POKE 123456,76     : REM INDIRIZ.
260 Y=256:  OPEN Y,3:  REM FILE NUMBER
270 Y=256:  A$=RIGHT$("A",Y):REM ARG.
```

ne segnalato nel caso in cui il numero di Data contenuti nelle apposite righe non sono sufficienti.

Quasi sempre, però, il motivo di tale segnalazione è dovuto ad un'errata trascrizione delle stesse linee: le cause più comuni sono la mancanza di una virgola (elemento separatore tra due Data), il posizionamento di una virgola in più, le virgolette (") che spesso sono troppe (o troppo poche).

Esaminate con cura il listato che contiene tutti i casi di errore più comuni.

Illegal Quantity

Questo tipo di errore, che conclude la rassegna di questo mese, è il più difficile da individuare perchè la causa è diversa a seconda del comando Basic che lo genera.

Di solito compare se, nella riga Basic, il valore non è presente in forma esplicita, ma associato ad una variabile numerica.

Il programma presenta una casistica piuttosto varia che segnala errore nel caso si assegni un valore negativo agli argomenti di Poke, Peek, Sqr, Log e così via; oppure si impone un valore troppo alto a Poke, Peek, Tab ed altri ancora.

Svilluppare una casistica completa è piuttosto arduo; si tenga presente, però, che Illegal vuol dire "Illegale, non consentito". Se dovesse capitarvi, quindi, chiedete subito al computer il valore delle variabili che sono la probabile causa dell'errore. Se, ad esempio, compare in una riga in cui è presente il comando...

Poke X,Sqr(Y)

...chidete il valore di X e di Y con un semplice...

Print X;Y; Sqr(Y)

I casi possono essere i seguenti:

- X è negativo oppure maggiore di 65535
- Y è negativo
- Sqr(Y) è un valore maggiore di 255.

Se controllate sul libretto di istruzioni del vostro computer, infatti, vi accorgerete che i valori indicati non possono essere elaborati dal calcolatore.

Dovete quindi "risalire" alle elaborazioni precedenti in cui una delle variabili (o entrambe) ha assunto, per errore, il valore inaccettabile; ed agire di conseguenza.

UNA, DIECI, CENTO ROUTINE

Programmare è facile e divertente; se, poi, esiste qualche routine che semplifica il lavoro, tanto meglio...

di **Alessandro Diano**

Vi sono giovanotti con meno di diciotto primavere i quali, dal loro hobby, riescono a tirar fuori un vero e proprio stipendio che conferma l'intelligenza (o quantomeno la furbizia) di un tale passatempo.

Purtroppo però, come spesso succede, il lavoro di creazione di un programma risulta parecchio dispendioso in termini di tempo e di energie, per la necessità di realizzare subroutine di Input controllato, o di animazioni Hi-Res e simili, le quali sono magari (o sicuramente?) già state affrontate da persone che, a loro volta, hanno imprecato per gli stessi problemi e, proprio su queste righe, cercano di presentarne i risultati, lasciando all'utente l'unica preoccupazione (peraltro non obbligatoria) di comprendere il perchè della subroutine stessa.

Quella che segue è una raccolta di esperienze che, in quanto tale, NON è possibile trovare altrove a causa della loro frammentarietà che, lungi dall'esser considerata negativa, è realmente valida ed utile a coloro che intendano far collezione di brevi ma utili nozioni di programmazione.

Queste pagine, insomma, rappresentano una sorta di "laboratorio informatico" al quale Commodore Computer Club si sforza sempre più di assomigliare.

UN BUON INPUT

Dando per scontato che in un programma di tipo professionale è d'obbligo l'impiego di una routine di Input è utile puntualizzare

Alcune Poke ed un paio di Sys possono realizzare veri e propri miracoli

Un Input controllato è piuttosto facile da strutturare

uno degli impieghi migliori dell'istruzione GET, regina incontrastata delle routine di Input controllato.

Se i tasti dei quali si attende la pressione hanno tutti codici contigui (esempio i CHR$ da 65 a 90, cioè l'intero alfabeto) il problema si risolve facilmente come in figura 1.

Se, invece, la richiesta deve prevedere, ad esempio, il riconoscimento delle sole vocali, di solito si ricorre alla spaghettata di IF...THEN per ciascuna delle possibili risposte, mentre si potrebbe risolvere lo stesso problema seguendo lo schema di fig. 2.

tervenire sulla locazione 646 relativa al colore dei caratteri) SOLO se, dopo aver impartito un comando di Print del colore del quale si desidera il cursore durante l'Input, si impartisce un CR$(147) per cancellare lo schermo e porre così il colore scelto come quello corrente.

A questo punto potranno essere adoperati altri colori, ma non un CHR$(147): basta che, durante l'Input, si modifichi il colore con quello prescelto all'inizio, oppure, molto più semplicemente, adoperare la Poke 207,0, come raccomandato, che funziona,

```
10 GET A$: IF A$<"A" OR A$>"Z" THEN 10
20 ON ASC(A$)-64 GOTO [A],[B],[C], ...

(Figura 1)
```

Il valore 5 presente in linea 10 indica la lunghezza della stringa inserita nella funzione MID$ (infatti "AEIOU" è lunga 5 caratteri), mentre le vocali tra parentesi in linea 20 andranno sostituite con i numeri di linea dai quali si vuole che il programma prosegua se vengono premuti i tasti "A", "E", "I", "O", oppure "U".

Sempre consigliabile, inoltre, l'uso delle Poke che abilitano un cursore artificiale durante il Get per le quali, ancora una volta, si riporta il programma di figura 3, stavolta comprensivo di TUTTE le Poke richieste.

Qui di seguito è presente il significato delle varie Poke:

- Poke 204,0 . Abilita il lampeggio del falso cursore
- Poke 205,20 . Regola la temporizzazione del lampeggio
- Poke 207,0 . Imposta il cursore su "lampeggio". ATTENZIONE! E' importante porre questa Poke tra il Get e l'IF...THEN per evitare che, durante l'Input, il cursore cambi di colore, assumendo quello corrispondente all'ultimo CHR$(147) impartito. In altri termini tale Poke può essere evitata (senza in- indipendentemente da quello corrente, qualunque colore venga assegnato al cursore in fase di Input.
- Poke 204,1. Disabilita il lampeggio del cursore
- Print CHR$(32). Stampa un carattere vuoto per eliminare l'eventuale spazio in reverse lasciato sullo schermo dal falso cursore.

Per completare la professionalità dell'Input, è inoltre consigliabile la Poke 198,0 prima di ogni Get o, ancora meglio, la routine di figura 4 che ha il pregio di essere compatibile con tutti i Commodore.

Ci si riferisce, chiaramente, alla linea numero 10 che "svuota" il buffer di tastiera da tutte le pressioni di tasti precedenti (volute od accidentali che siano) e che può benissimo essere completata da una successiva linea di Return che ne consenta l'uso con un semplice Gosub.

Un'altra considerazione, particolarmente preziosa, è quella di impedire al programma di cacciarsi nei guai se qualche utente maligno cerca di chiedere dati da periferiche non connesse, allo scopo di farlo impal-

```
10 GET A$: A=1: FOR B=1 TO 5: IF A$=MID$("AEIOU",B,1) THEN A=B+1
20 NEXT B: ON A GOTO 10,[A],[E],[I],[O],[U]

(Figura 2)
```

lare. Premesso che nell'I/O la gestione errori dev'essere totale, è doveroso far girare il programma con la tastiera completamente disabilitata (oltre, chiaramente, al Run/Stop e Restore, Reset eccetera), servendosi di una Poke 649,0 posta in una delle prime linee, ed agendo, come indicato in figura 5, quando serve un ingresso dati.

In pratica si abilita la tastiera solamente quando è necessario; al limite, nella linea 10, se si attende un solo carattere si può anche ricorrere ad una Poke 649,1: così facendo si evita che una piccola distrazione in fase di previsione errori non rovini un buon lavoro di programmazione.

```
10 PRINT "Scelta? ";: POKE 204,0: POKE 205,20: POKE 198,0
20 GET A$: POKE 207,0: IF A$="" THEN 20
30 IF ASC(A$)<>13 THEN PRINT A$;: GOTO 20
40 POKE 204,1: PRINT CHR$(32)
```

(Figura 3)

DATA ERROR? NO GRAZIE!

Nelle mappe di memoria per il Commodore 64, si legge che le locazioni 63 ($3F) e 64 ($40) contengono il numero della linea Data corrente.

La faccenda può risultare utile in svariate situazioni; una delle più tipiche è la seguente: durante una lettura di Data, il computer può segnalare un "Illegal quantity error" nella linea...

For A=49152 to 50000: Read B: Poke A,B: Next A

E' chiaro che l'errore sta nella variabile B che, molto probabilmente, conterrà un valore "illegale" che, nel caso della Poke, sarà maggiore di 255: ma quale linea Data contiene tale valore?

La risposta sarà fornita dal comando...

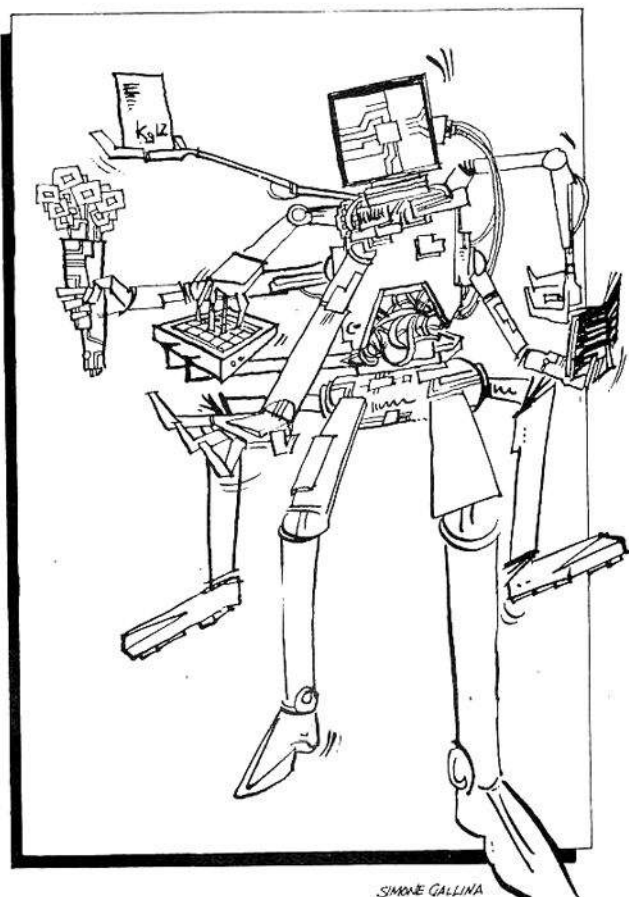
Print 256 * Peek(64) + Peek(63)

...il quale, sebbene introdotto in tal caso in modo diretto, può essere inserito nel ciclo di For... Read... Next per rivelare la linea percorsa dal S.O. durante l'esecuzione di Read.

C'E' SPAZIO IN MEMORIA?

Il comando Print FRE(0) serve per fornire all'utente un'indicazione della memoria Basic libera rimasta a disposizione, anche se è

Ecco come individuare la riga Data appena esaminata dal computer

SIMONE GALLINA

Stampare un messaggio in l.m. è facile quasi come in Basic

```
10 GET A$: IF A$<>"" THEN 10
20 GET A$: IF A$="" THEN 20
30 ON ASC(A$+CHR$(0)) GOSUB .... etc.
```

(Figura 4)

spesso più utile sapere quanto spazio il programma occupa. Onde evitare risposte di valori negativi, è consigliabile la seguente sintassi del comando:
Print FRE(0)+65536 per la memoria rimasta libera;
Print -FRE(0)-26625 per la memoria occupata dal programma

CONTRO IL LOAD ERROR

Se si è affezionati utenti del registratore a cassette, si avrà anche una certa familiarità con l'antipatico messaggio di "Load error". Questo, comunque, può essere ignorato se, alla ricomparsa del cursore, dando un LIST non si nota alcuna "sporcizia" nel listato: se questo appare perfettamente integro, si può introdurre in forma diretta il comando:
Poke 45,Peek(831): Poke 46,Peek(832): CLR

```
10 POKE 649,10
20 [input richiesto]
30 POKE 649,0
```

(Figura 5)

La stessa sequenza di comandi viene impiegata per caricare un programma in metà tempo, schiacciando il Run/Stop a metà del caricamento, a patto, ovviamente, di saperlo fin dall'inizio, e digitando, appunto, quanto sopra. Può addirittura essere inserito come prima linea di un programma, quale forma di protezione da un eventuale "Load error". Se il LOAD avviene da disco, chiaramente, quanto esposto sopra NON risulta essere assolutamente applicabile.

L'ALTER-EGO DEL JOYSTICK

Se non possedete il joy potete usare la tabella di equivalenza indicata in figura 6 che, tuttavia, non sempre è valida con i programmi commerciali.

SCREEN EDITOR

Soprattutto nella schermata di presentazione di un programma si sente la necessità di disporre di funzioni particolari che permettano la manipolazione delle singole linee dello schermo, allo scopo di inserire qualche novità di visualizzazione. Se opportunamente combinate, le seguenti istruzioni possono davvero contribuire alla creazione di effetti altrimenti irrealizzabili dai consueti cicli For / Next:

FUNZIONE	PORTA 1	PORTA 2
Alto	"1"	Spazio + F1
Basso	Freccia a Sx.	Spazio + "Z"
Destra	"2"	Spazio + "B"
Sinistra	Control	Spazio + "C"
Fuoco	Spazio	Spazio + "M"

(Figura 6)

```
1000 48        PHA           Salva l'accumulatore nello stack
1001 8A        TXA           Trasferisce il registro X nell'accumulatore
1002 48        PHA           Salva il registro X nello stack
1003 98        TYA           Trasferisce il registro Y nell'accumulatore
1004 48        PHA           Salva il registro Y nello stack
1005 A0 00     LDY #$00      Y=0
1007 BA        TSX           Copia il puntatore di Stack nel registro X
1008 FE 04 01  INC $0104,X   Incrementa il byte basso dell'indirizzo di RTS
100B D0 03     BNE $1010     Salta se non vi e' overflow
100D FE 05 01  INC $0105,X   Incrementa il byte alto dell'indirizzo di RTS
1010 BD 04 01  LDA $0104,X   Carica il byte basso dell'indirizzo di RTS
1013 85 35     STA $35       Lo pone in pagina zero
1015 BD 05 01  LDA $0105,X   Carica il byte alto dell'indirizzo di RTS
1018 85 36     STA $36       Lo pone in pagina zero
101A B1 35     LDA ($35),Y   Carica il byte "puntato" da $35/$36
101C F0 05     BEQ $1023     Esce se e' uguale a zero
101E 20 D2 FF  JSR $FFD2     Altrimenti lo invia al corrente device d'output
1021 90 E4     BCC $1007     Se non vi sono errori prosegue col prossimo carattere
1023 68        PLA           Richiama il valore del registro Y
1024 A8        TAY           Lo ripristina
1025 68        PLA           Richiama il valore del registro X
1026 AA        TAX           Lo ripristina
1027 68        PLA           Richiama il valore dell'accumulatore
1028 60        RTS           Termine della routine, ritorna al programma
                             principale
```

(Figura 7)

• Poke 781,n: SYS 59903 cancella il contenuto della sola riga video "n" lasciando immutate le altre.
• SYS 59626 trasferisce il contenuto di ciascuna riga in quella immediatamente superiore, "scrollando" in pratica il video verso l'alto di una posizione.

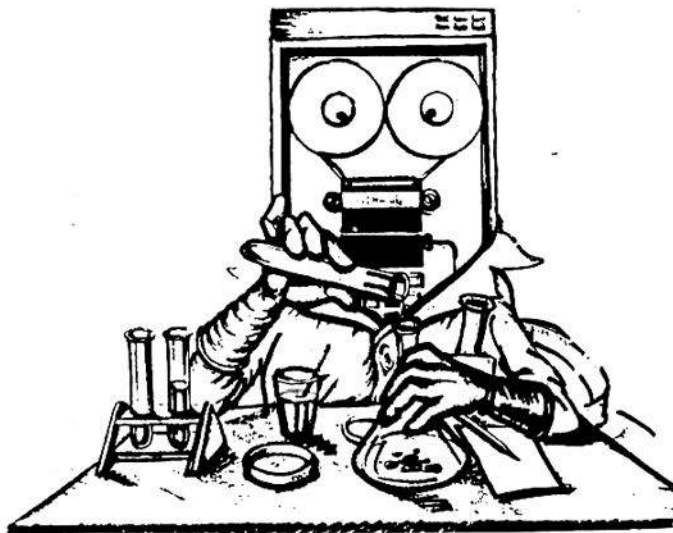
Esiste anche la possibilità di fare copie di linee in maniera ancor più generale: se L1 è la linea da copiare ed L2 è la linea nella quale dev'essere copiata L1, è possibile adoperare le seguenti istruzioni:

10 Poke 781,L1: SYS 59888: Poke 172, Peek(60656+L2)
20 Poke 780,Peek(216+L2): SYS 59848

La faccenda è di chiara derivazione Assembly, come rivelano le due Poke relative alle locazioni 780 e 781 che interessano, rispettivamente, l'accumulatore ed il registro X.

PRINT IN ASSEMBLY

In l.m. la stampa di messaggi non è altrettanto immediata come in Basic, dove basta far seguire all'apposita istruzione (Print) l'insieme dei caratteri da visualizzare; solitamente si carica uno dei registri con il numero dei caratteri da stampare e si opera una serie di LDA indicizzati seguiti da altrettanti JSR alla mitica subroutine di CHROUT posta a $FFD2.

Più volte è stato riproposto il problema di disporre di una JSR che equivalesse ad un comando di Print, nel senso che stampasse i caratteri successivi la JSR stessa, ed evidentemente la questione è arrivata sino alla stessa Commodore dal momento che le macchine ad otto bit successive al C/64, tra le altre novità, presentano, nella Jump

	Jump Table	Inizio routine
C/16 & Plus/4	$FF4F	$FBD8
Commodore 128	$FF7D	$FA17

(Figura 8)

Table, anche un JMP d'accesso alla routine di PRIMM la cui funzione è proprio quella richiesta: invia alla periferica d'uscita corrente (la numero tre, di default, corrispondente al video) i valori che seguono la JSR sino a trovare uno zero.

I vantaggi sono evidenti: non è necessario destreggiarsi tra LDX, INX e compagnia varia, ma è sufficiente un semplice JSR PRIMM, seguito dalle scritte volute, concluse da un byte nullo, dopo il quale riprende il disassemblato dell'utente.

Tra gli svantaggi, però, si nota l'obbligo di traslare l'intero programma in caso di modifica delle scritte (mentre nel metodo tradizionale lo si poteva evitare posizionandole in zone differenti da quella del programma) e la scarsa leggibilità che contraddistingue un disassemblato facente uso della PRIMM.

In ogni caso è notevole la comodità che una tale routine consente di avere nella gestione dell'output su video e si ritiene opportuno fornire il disassemblato della versione ottimizzata (con lievi modifiche) tratta dal S.O. del C/128; quella del C/16 e Plus/4, pur fornendo le medesime prestazioni, è infatti più lunga e confusa a causa della presenza di inutili LDX ed INX che sono infatti stati eliminati nella nuova versione, sostituiti dall'incremento diretto del valore nello stack (INC $104,X).

La preziosa routine, con i relativi commenti, è rappresentata dalla figura 7.

Indubbiamente i signori Bowen, Ertwine e Ryan (autori dell'originale) hanno creato una routine semplice quanto potente che risulta essere pulita da sfronzoli inutili. Due solamente le possibili indicazioni: il puntatore scelto in pagina zero ($35 / $36) viene impiegato, nel Commodore 64, per funzioni di utilità inerenti proprio le stringhe ed è sicuramente meno sfruttato dei suoi colleghi $FB / $FC e $FD / $FE i quali, di solito, in un programma Assembly sono sempre oberati di lavoro.

La seconda osservazione riguarda la caratteristica di rilocabilità della routine poichè è possibile un suo impiego a partire da una locazione diversa (da $1000 nell'esempio riportato) operando con un banale comando di Transfer senza problema alcuno.

Chi possiede il C/128 (oppure il C/16 o Plus/4) può usare il Monitor incorporato per esaminare la ROM del computer per conoscere la PRIMM che i rispettivi S.O. hanno già implementata (figura 8).

Alcune operazioni importanti del C/64 sono riportate nella figura 9, relativa alle Po-

```
Inizio della pagina grafica:

POKE 53272,(PEEK(53272) AND 240) OR N

                     Posizione della Pag.Graf.
Valore di  N       Decimale       Esadecimale
      0                0              $0000
      8             8192              $2000

(Figura 9)
```

```
Abilita la pagina grafica standard:

POKE 53265,PEEK(53265) OR 32


Abilita la pagina grafica multicolore:

POKE 53265,PEEK(53265) OR 32
POKE 53270,PEEK(53270) OR 16


Disabilita la pagina grafica standard:

POKE 53265,PEEK(53265) AND 223


Disabilita la pagina grafica multicolore:

POKE 53265,PEEK(53265) AND 223
POKE 53270,PEEK(53270) AND 239


Seleziona il banco video:

POKE 56576,(PEEK(56576) AND 252) OR N

Valore di  N    Nr. Banco   Indirizzo esa
     0              3           $C000
     1              2           $8000
     2              1           $4000
     3              0           $0000


(Figura 10)
```

ke per la selezione della pagina grafica all'interno di uno dei quattro banchi selezionabili.

La figura 10, invece, seleziona uno dei 4 banchi video disponibili.

La figura 11, relativa alla mappa dei caratteri, non ha bisogno di commenti, e così pure la 12, relativa alla selezione della memoria di schermo all'interno dei 16k "visti" dal chip video.

Alcune locazioni di particolare interesse sono riportate qui di seguto:

197 Lettura della tastiera
646 Colore dei caratteri
53280 Colore della cornice
53281 Colore dello sfondo
54297 Lettura della paddle 1
54298 Lettura della paddle 2
56320 Lettura joystick 1 e pulsante paddle
56321 Lettura joystick 2 e pulsante paddle
56325 Velocità del cursore (0=veloce)

In figura 13, infine, riportiamo alcune Poke di più frequente impiego.

Seleziona la memoria per i caratteri:

```
POKE 53272,(PEEK(53272) AND 240) OR N
```

Valore di N	Posiz. Mappa Caratt. Decimale	Esadecimale
0	0	$0000
2	2048	$0800
4	4096	$1000
6	6144	$1800
8	8192	$2000
10	10240	$2800
12	12288	$3000
14	14336	$3800

(Figura 11)

Seleziona la memoria per lo schermo:

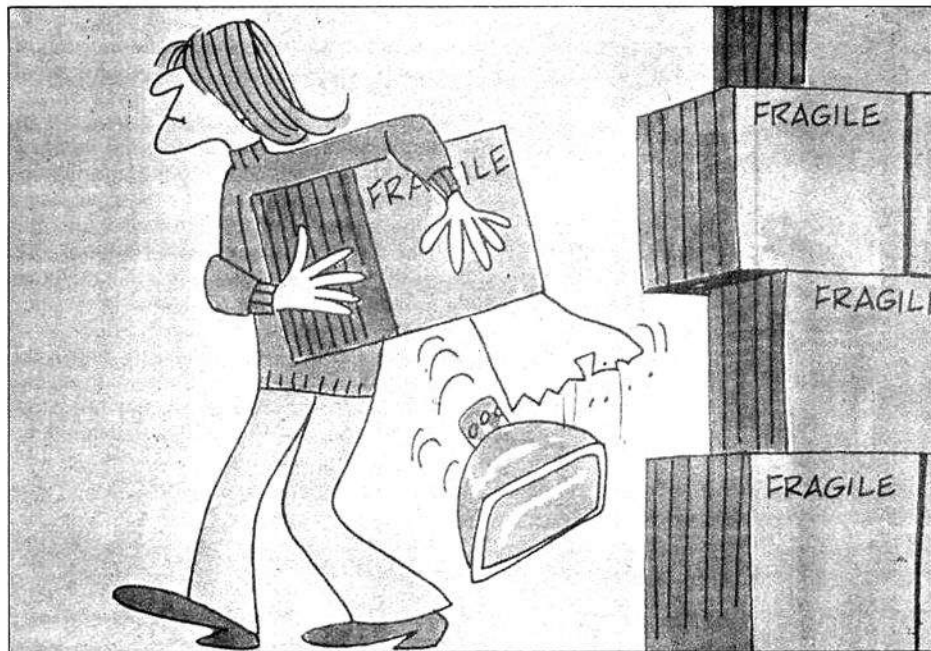
```
POKE 53272,(PEEK(53272) AND 15) OR N
```

Valore di N	Posizione della M.S. Decimale	Esadecimale
0	0	$0000
16	1024	$0400
32	2048	$0800
48	3072	$0C00
64	4096	$1000
80	5120	$1400
96	6144	$1800
112	7168	$1C00
128	8192	$2000
144	9216	$2400
160	10240	$2800
176	11264	$2C00
192	12288	$3000
208	13312	$3400
224	14336	$3800
240	15360	$3C00

(Figura 12)

Poke principali:

Funzione	Disabilita	Riabilita
N.ri di linea	22,35	22,25
Tastiera	649,0	649,10
Ripetiz. tasti	650,128	650,0
LIST	775,191	775,167
Restore	792,193	792,71
Run/Stop	808,239	808,237
LOAD	816,157	816,165
SAVE	819,246	819,245

(Figura 13)

UNO SCHERMO DI SCORTA

Sfruttiamo la velocità supersonica dell'Assembly per disporre, anche nei nostri programmi Basic, di videate in tempo reale

di **Domenico Pavone**

Nella programmazione in Basic, uno degli aspetti più ardui da gestire è senza dubbio quello che riguarda il trattamento, e quindi anche il trasferimento, di vaste aree di memoria.

Questo non perchè il problema presenti grosse difficoltà teoriche: sappiamo tutti che una manciata di Poke, inserita in un ciclo For...Next, può fare miracoli; ma la lentezza del Basic impedisce rapidità soddisfacenti.

Spesso qualche secondo di attesa può essere giustificato in applicazioni senza pretese estetiche, ma è anche vero che talvolta il Basic diventa improponibile, e comunque neanche lontanamente paragonabile all'efficacia dell'Assembly.

Per non restare sul vago, prendiamo in considerazione la manipolazione di intere schermate grafiche, limitandoci per ora a quelle in bassa risoluzione; ovvero a quell'area di memoria del C/64 predisposta a visualizzare sullo schermo tutto ciò che si digita alla tastiera.

Questo consentirà di avvicinarsi ulteriormente alle tecniche di programmazione in Assembly, prendendo confidenza con alcuni dei più comuni codici mnemonici di questo linguaggio.

E' facile imparare il l.m. soprattutto utilizzando le varie istruzioni in diverse applicazioni

Come procedere

Vediamo, prima di tutto, di definire chiaramente che cosa ci proponiamo di fare.

Com'è noto, il C/64 consente di visualizzare sullo schermo 40x25 = 1000 caratteri, per cui esiste in memoria un'area adibita

Un qualsiasi Monitor l.m. è utilizzabile per disassemblare la routine proposta

a tale scopo che, per forza di cose, sarà composta di 1000 locazioni di memoria. Altrettanto noto è che tali locazioni sono quelle comprese da 1024 a 2023, e che, "pokandovi" opportuni valori numerici, compaiono lettere, simboli, eccetera.

Invertendo il procedimento, se si digita una serie di caratteri alfanumerici o semigrafici (= quelli ottenibili premendo SHIFT o il tasto Commodore), i loro equivalenti codici di schermo possono essere "estratti" dalla suddetta area di memoria mediante delle comuni Peek.

Date queste (scontate) premesse, supponiamo ora di volere che un nostro programma, in uno o più momenti della sua esecuzione, mostri una certa videata molto complessa, magari un "fantamessaggio" di copyright col nostro nome ben contornato da cuoricini, linee e pallini vari che riempiono tutto lo schermo.

Scartata l'ipotesi di imbottire il programma con centinaia di "Print" diverse, un metodo più pratico potrebbe essere sintetizzato così:

- Preparare una schermata, con tutta calma, e sfruttando le facilitazioni fornite dal normale editor (uso del cursore, del delete, ecc.).
- Leggere, tramite 1000 Peek, il contenuto delle locazioni di memoria da 1024 a 2023, e depositarle mediante Poke in un'area di memoria "tranquilla", come quella presente da 49152 in poi, o che non subisca interferenze dal Basic.
- Ogni qual volta si vuole fare apparire la schermata, che rimane inalterata, si procede in senso inverso rispetto alla seconda fase, leggendo le 1000 locazioni da 49152 a 50151 e pokandole nella solita area video.

CARTA D'IDENTITA'

Registro: ACCUMULATORE

Stato di famiglia: 65XX

Residenza abituale: CPU

Dimensioni: 8 BIT (1 BYTE)

Professione: Utilizzando gli opportuni codici dell'Assembly, è possibile effettuare sul suo contenuto svariate operazioni, che vanno dal semplice "scaricamento" in una certa locazione di memoria (STA), fino all'esame dei suoi singoli bit. Per sfruttare appieno le possibilità del registro accumulatore, è comunque necessario applicarsi con costanza nell'apprendimento dell' Assembly.

Segni particolari: è l'unità interna della CPU più versatile nella manipolazione dei dati.

Caratteristiche fisiche: trattandosi di una struttura ad 8 bit, al suo interno possono essere immagazzinati solo valori compresi tra 0 e 255. A differenza dei suoi cugini "X" ed "Y", è fisicamente connesso all'unità di calcolo del microprocessore, la ALU (Unità Logico Aritmetica); il che consente di effettuare, sul valore in esso immagazzinato, operazioni di somma (ADC) e sottrazione (SBC): il risultato di tali operazioni viene depositato nell'Accumulatore stesso, sostituendosi al dato prima contenuto, che viene quindi perso irrimediabilmente.

L'altissima velocità operativa del l.m. pone in secondo piano la lentezza del Basic

CARTA D'IDENTITA'

Registri: X ed Y

Residenza: CPU

Dimensioni: 8 BIT (1 BYTE)

Stato di famiglia: entrambi i registri possono rapportarsi direttamente con l'Accumulatore mediante istruzioni che consentono un velocissimo trasferimento del loro contenuto da e verso quel registro. Per esempio, TXA copierà nell'accumulatore il contenuto di X, mentre TAX compirà il lavoro opposto (lo stesso vale per TYA e TAY, che riguarderanno il registro Y). Non esiste, nel set di istruzioni del 6510, un codice di trasferimento diretto tra X e Y.

Professione: di impiego praticamente analogo a quello dell'accumulatore, vengono per lo più utilizzati per memorizzazioni temporanee di dati ed indicizzazioni varie.

Caratteristiche fisiche: come l'Accumulatore, possono essere "caricati" (LDX, LDY) con dati numerici compresi tra 0 e 255, sui quali però si può agire direttamente solo incrementandone o decrementandone il valore di una unità (INX, DEX, INY, DEY). Questa peculiarità viene di solito sfruttata utilizzando i due registri come contatori per cicli iterativi tipo il For...Next del Basic.

Come anticipato all'inizio, si tratta semplicemente di trasferire una porzione di memoria da una zona ad un'altra. Ma andiamo a vedere in concreto come lavora il nostro computer, cominciando col dare un'occhiata al listato 1, da digitare e salvare su disco o su nastro.

In Basic

Le prime righe del programma (40 - 60) sono, per così dire, un optional, in quanto si limitano a darvi una mano per creare una schermata personalizzata, ricorrendo a un vecchio trucco Basic, spesso usato per non far apparire il punto interrogativo quando si usa l'istruzione INPUT. In pratica, dopo il RUN, potrete disegnare o scrivere (nonchè cancellare) sullo schermo tutto ciò che volete, badando solo a non usare il tasto Return, se non per uscire da questa fase, che potremmo definire di "mini editing".

Alla pressione del tasto Return, infatti, e dopo un'attesa di parecchi secondi, il programma continuerà la sua esecuzione uscendo dall'istruzione di linea 55. Una ulteriore facilitazione potrebbe venirvi dalla ripetizione del tasto premuto, che è possibile attivare aggiungendo, alla linea 45, il comando Poke 650,128.

Se il vostro estro creativo non ha voglia di emergere, potete cancellare le righe in questione ed utilizzare una linea tipo...

```
60 For X=1024 To 2023: Poke X,80:
Next
```

Con le righe di istruzioni 80 e 85 si attua il trasferimento vero e proprio dell'area video copiandola nelle locazioni da 49152 in poi, mentre con le linee 130 e 135 si effettua il compito inverso.

Mandando in esecuzione il programma, e premendo un tasto dopo la comparsa della relativa segnalazione, potrete constatare come alla sua notevole semplicità non corrisponda una effettiva praticità d'uso, a cau-

E' sufficiente cambiare i byte di "puntamento" per trasferire su video una qualsiasi porzione di 1000 byte di memoria

sa della notevole lentezza con cui avviene il trasferimento, chiaramente percepibile nella fase di "ritorno" dell'immagine memorizzata.

Se, però, affidiamo questo compito ad una routine in linguaggio macchina, ecco che i vantaggi di un simile modo di procedere diventeranno palesi, e tali da giustificarne l'impiego in più di una applicazione.

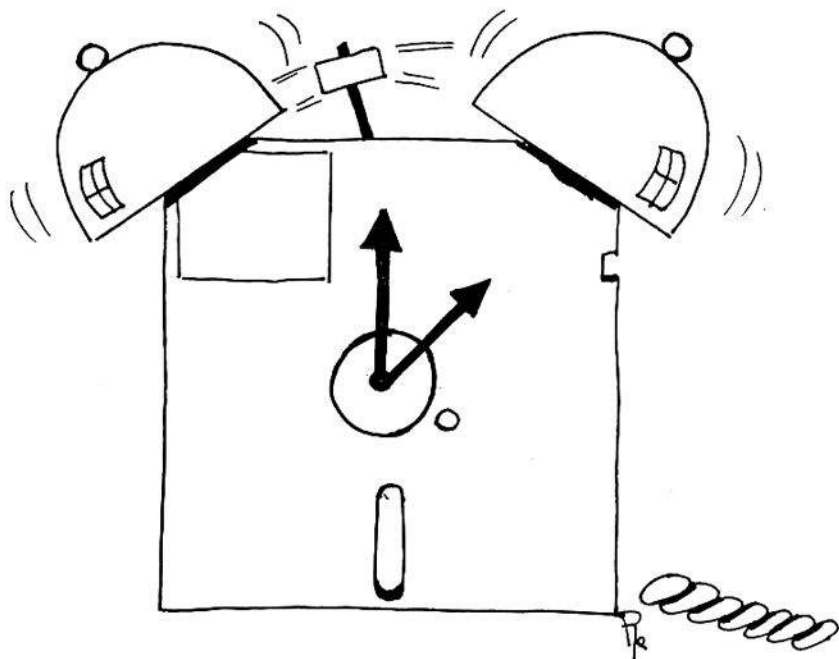
Un fulmine chiamato Assembly

Come ormai dovreste sapere, il primo passo per sfruttare le reali capacità di elaborazione del C/64 consiste nella creazione e allocazione in memoria di una routine che operi in linguaggio macchina, attivabile poi da Basic tramite la SYS opportuna.

Per raggiungere lo scopo, cominciate col copiare il breve listato 2, badando a non commettere errori nella digitazione delle linee Data. Questo tipo di programmi vengono di solito definiti col termine "caricatori Basic" per la funzione che essi svolgono, quella appunto di "caricare" in memoria il codice macchina (numerico) di una routine Assembly: nel nostro caso, la routine di trasferimento illustrata dal disassemblato, che potete vedere a parte, e che esamineremo dettagliatamente tra breve.

Impartite il RUN e, se tutto è in ordine (in caso contrario apparirà una segnalazione di errore), ricomparirà il solito READY e sembrerà (attenzione: SEMBRERA') che nulla sia accaduto. Il programma LM, invece, è ora allocato in memoria nelle locazioni che vanno da 50176 in poi, in modo da non interferire con l'area che verrà usata come deposito (buffer) per la copia della nostra schermata (locazioni da 49152 a 50151).

Ora, senza spegnere il computer, ricaricate il programma del listato 1 e modificatelo in maniera che risulti uguale al listato 3 (o digitatelo ex novo). Salvate anche questo, e finalmente attivatelo col solito RUN.

Stavolta, dopo aver creato la schermata, la segnalazione di trasferimento avvenuto apparirà quasi contemporaneamente alla pressione del tasto Return, così come la ricomparsa dell'immagine "archiviata". Questa, tra l'altro, può essere richiamata tutte le volte che si vuole; provate infatti a digitare RUN 130 (oppure: Goto 130), e la vedrete riapparire in un baleno. Se non si alterano le locazioni da 49152 in poi con Poke varie, la schermata (e il programma LM) rimarrà a vostra disposizione fino allo spegnimento del computer, resistendo anche a eventuali reset.

Come funziona

Non ci resta che esaminare come la routine di trasferimento dell'area video svolga il suo compito.

Facendo riferimento al disassemblato in figura, seguiamone il flusso operativo, istruzione per istruzione.

Le prime righe (13 - 15) costituiscono la fase di inizializzazione, in cui si assegnano ad alcune etichette valori precisi, in modo che, in seguito, l'assembler sappia "riconoscerle". PART (= partenza) corrisponderà quindi alla locazione 1024 (l'inizio dell'area video) e ARR (= arrivo) a 49152 (inizio del buffer).

Con la riga 15 si specifica, tramite il simbolo "*=", la locazione iniziale in cui si intende allocare la routine, in questo caso da 50176 in poi.

Ultimati i preparativi, inizia la vera e propria esecuzione, che con le linee da 17 a 19 "poka" il valore 0 in un non meglio (per ora) precisato VID+1 e BUF+1. E' questo il nodo cruciale da capire, per cui fermiamoci per approfondire la correlazione esistente tra i codici mnemonici ed i loro corrispettivi numerici, vale a dire il codice LM.

Scorrendo il disassemblato vediamo che le etichette VID e BUF si trovano alle righe 28 e 29, le quali effettuano il vero e proprio trasferimento, caricando in accumulatore il contenuto della locazione di partenza (PAR) e copiandolo nella locazione di arrivo (ARR). Per capire bene cosa identificano VID+1 e BUF+1, fate ricorso all'uso di un qualsiasi monitor, naturalmente dopo aver allocato in memoria la routine LM.

Supponendo di utilizzare l'ottimo "Macro Assembler" Commodore, attivate il programma 2; caricate quindi il Monitor $8000 (contenuto nel dischetto originale) e attivatelo con SYS 32768. Digitate ora...

```
D C400 C429
```

...che sta per: Disassembla (D) le locazioni da 50176 (esadec.= C400) a 50125 (esadec.= C429).

Vedrete apparire le stesse istruzioni del disassemblato MacroAssembler, ma stavolta in un formato privo di etichette, sostituite negli operandi dai relativi indirizzi, e con il reale posizionamento della routine al posto dei numeri di linea.

In particolare, quelle che erano le righe 28 e 29 appariranno ora nel modo seguente:

```
C413  AD 00 04 LDA $0400
C416  8D 00 C0 STA $C000
```

Il primo valore (C413 = dec. 50195) indica, in esadecimale, la posizione in memoria dell'istruzione LDA, e quindi il reale indirizzo dell'etichetta VID, così come C416 (dec. = 50198) corrisponderà a BUF. I tre valori che seguono identificano nell'ordine: l'istruzione (1 byte) e l'indirizzo. Quest'ultimo, viene immagazzinato in memoria nel formato byte basso / byte alto, per cui 00 04 corrisponderà all'indirizzo 0400 (dec=1024) e 00 C0 a C000 (dec 49152).

In pratica, VID+1 corrisponde alla locazione 50196, e quindi al byte basso dell'indirizzo di partenza, mentre VID+2 ne indicherà il byte alto. Lo stesso vale ovviamente per BUF+1 e BUF+2.

Chiusa questa parentesi, torniamo al nostro disassemblato.

Le righe da 20 a 25 sfruttano i registri "contatori" X ed Y per creare un ciclo iterativo che ripeta 1000 volte l'istruzione di trasferimento di una locazione. Dato che infatti non è possibile caricare in essi un valore maggiore di 255, si decrementa il registro

Il registro X rappresenta un validissimo contatore piuttosto versatile

"X" per 250 volte finchè non è eguale a zero, e quello "Y" quattro volte, per cui 250x4=1000.

L'istruzione BNE (Branch if Not Equal) indica un salto condizionato, e va interpretata così: "Se l'ultima operazione svolta non ha dato come risultato 0, allora salta all'indirizzo (nel nostro caso specificato da una label) indicato dall'operando, altrimenti prosegui con la successiva istruzione".

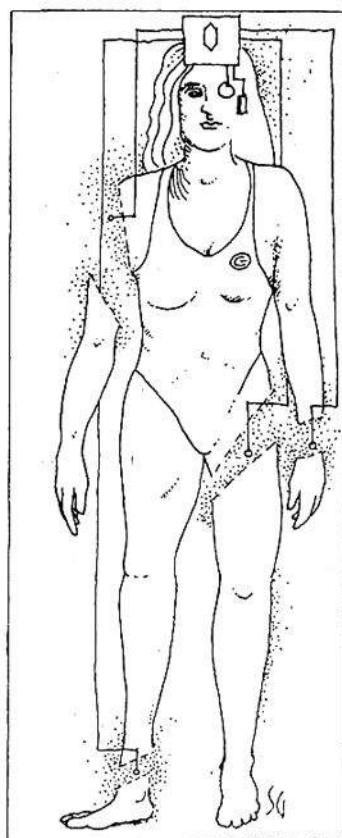
Ogni volta che viene effettuato un trasferimento, occorre ovviamente incrementare l'indirizzo di partenza e di arrivo, e di ciò se ne occupano le linee da 31 a 36, incrementandone il byte basso (VID+1 e BUF+1) finchè è diverso da zero, cioè finchè non raggiunge il valore 255, altrimenti incrementa il byte alto (VID+2 e BUF+2). A questo proposito è opportuno ricordare che, in Assembly, incrementando oltre 255 il valore di un registro (o di una locazione), si torna al valore 0, così come decrementando 0 si torna a 255.

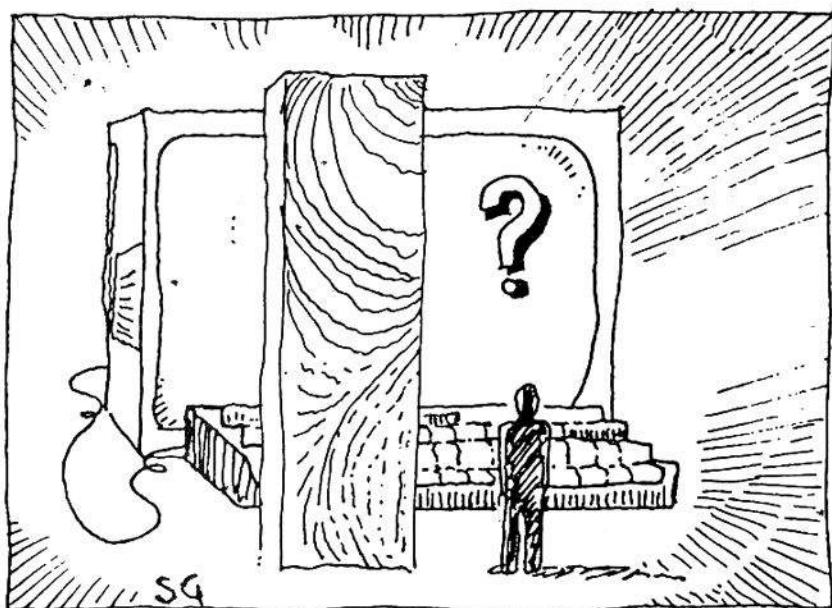
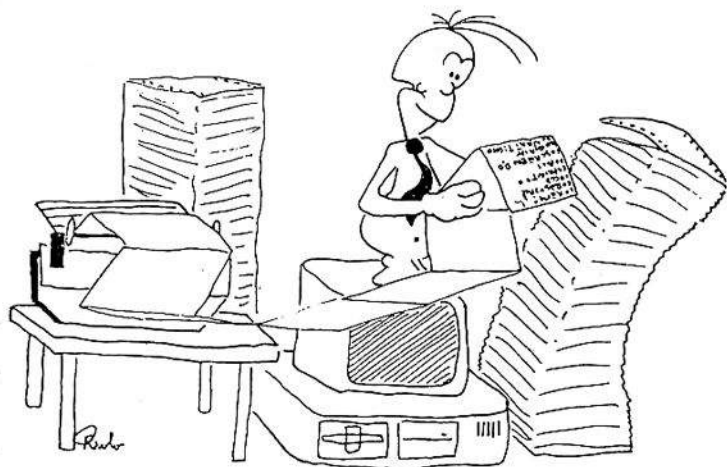
L'istruzione di riga 36 (JMP = Jump) è in tutto e per tutto identica al GOTO del Basic, mentre RTS (= ReTurn from Subroutine) di linea 26 consente, a operazioni ultimate, di tornare al Basic.

Sulla base di queste precisazioni, è possibile ora capire come la routine possa agire sia per immagazzinare una schermata, che per riposizionarla nella normale area video: è sufficiente, infatti, invertire i valori di PART e ARR per mutare la direzione del trasferimento, ed è proprio questo il compito delle Poke che appaiono nel listato 3 (righe 85 e 130) prima della SYS che attiva la routine LM.

Non resta da aggiungere che quello illustrato non è che uno dei modi possibili per trasferire vaste aree di memoria da una zona all'altra. Sfruttando appieno le capacità del Macro Assembler, e con una certa costanza di applicazione, già da una breve routine come questa è possibile tirarne fuori gli usi più disparati, limitati solo dall'inventiva del programmatore.

E qui entrate in ballo voi.

```
10 REM -------------------------
15 REM       L I S T A T O  1
20 REM -------------------------
25 :
30 REM ***  EDITOR SCHERMATA  ***
35 :
40 PRINTCHR$(147):PRINT"CREA UNA";
45 PRINT" VIDEATA E PREMI RETURN"
50 OPEN1,0: REM ---> APRE FILE SU
55 INPUT#1,A$:REM -> PERIFERICA 0
60 CLOSE1: REM ---->  (TASTIERA)
65 :
70 REM *** SCHERMO IN BUFFER  ***
75 :
80 FORX=0TO999
85 POKE49152+X,PEEK(1024+X):NEXT
90 PRINTCHR$(147)
95 PRINT"SCHERMATA  MEMORIZZATA"
100 PRINT:PRINT"PREMI UN TASTO ";
105 PRINT"PER RICARICARLA"
110 GETX$:IFX$=""THEN110
115 :
120 REM *** BUFFER IN SCHERMO ***
125 :
130 PRINTCHR$(147):FORX=0TO999
135 POKE1024+X,PEEK(49152+X):NEXT
```

```
10 REM --------------------
20 REM       LISTATO  2
30 REM --------------------
35 :
40 FORX=0TO41:READA:B=B+A
50 POKE 50176+X,A:NEXT
60 IFB=5626THENEND
70 PRINT"ERRORE NEI DATA!"
75 :
110 DATA 169,000,141,020,196
120 DATA 141,023,196,160,004
130 DATA 162,251,202,208,004
135 DATA 136,208,248,096,173
140 DATA 000,004,141,000,192
150 DATA 238,020,196,238,023
155 DATA 196,208,235,238,021
160 DATA 196,238,024,196,076
165 DATA 012,196
```

```
10 REM -------------------------
15 REM       L I S T A T O  3
20 REM -------------------------
25 :
30 REM ***  EDITOR SCHERMATA   ***
35 :
40 PRINTCHR$(147):PRINT"CREA UNA";
45 PRINT" VIDEATA E PREMI RETURN"
50 OPEN1,0: REM ---> APRE FILE SU
55 INPUT#1,A$:REM -> PERIFERICA 0
60 CLOSE1: REM ---->  (TASTIERA)
65 :
70 REM *** SCHERMO IN BUFFER   ***
75 :
85 POKE50197,4:POKE50200,192
90 SYS50176:PRINTCHR$(147)
95 PRINT"SCHERMATA  MEMORIZZATA"
100 PRINT:PRINT"PREMI UN TASTO ";
105 PRINT"PER RICARICARLA"
110 GETX$:IFX$=""THEN110
115 :
120 REM *** BUFFER IN SCHERMO ***
125 :
130 POKE50197,192:POKE50200,4
135 SYS50176
```

```
-------------------------------------------
         DISASSEMBLATO MACROASSEMBLER
-------------------------------------------
10 ;
11 ;  TRASFERIMENTO AREA VIDEO
12 ;
13 PAR     =$0400      ;Decimale = 1024
14 ARR     =$C000      ;Decimale = 49152
15         *=$C400     ;Decimale = 50176
16 ;----------------------------------------
17         LDA #00     ;Azzera byte basso
18         STA VID+1   ;indir. partenza
19         STA BUF+1   ;e arrivo.
19 ;----------------------------------------
20         LDY #04     ;Y = 4
21 START   LDX #251    ;X = 251
22 CICLOX  DEX         ;X = X - 1
23         BNE VID     ;Se X <> 0 goto 28
24 CICLOY  DEY         ;Y = Y - 1
25         BNE START   ;Se Y <> 0 goto 21
26         RTS         ;Ritorna al basic.
27 ;----------------------------------------
28 VID     LDA PAR     ;A = peek (PART)
29 BUF     STA ARR     ;Poke ARR, A
30 ;---------- ----------------------------
31         INC VID+1   ;Inc.di 1 low byte
32         INC BUF+1   ;part. e arrivo.
33         BNE CICLOX  ;Se <> 0 salta.
34         INC VID+2   ;Inc.di 1 hi byte
35         INC BUF+2   ;part. e arrivo.
36         JMP CICLOX  ;Goto 22
```

ALLA SCOPERTA DEL MACRO ASSEMBLER

Il "Macro Assembler" della Commodore è un package sicuramente utile per chi voglia affrontare seriamente, con il C/64, l'affascinante mondo del linguaggio macchina

di **Domenico Pavone**

Nel dischetto del Macro Assembler sono presenti due programmi di monitor...

Monitor $8000
Monitor $C000

...che, caricati con la sintassi...

LOAD "....",8,1

...vengono automaticamente collocati in memoria, il primo a partire dalla locazione 32768 (in esadecimale: $8000), il secondo da 49152 in poi (esadecimale = $C000).

La scelta su quale dei due utilizzare dipende dall'area di memoria sulla quale si intende operare: se, per esempio, vogliamo intervenire su una routine allocata da 49152 in poi, dovremo obbligatoriamente ricorrere al "Monitor $8000", in quanto l'altro si sovrapporrebbe al programma da esaminare, distruggendone il contenuto.

Come è ovvio, per attivare Monitor $8000 occorrerà impartire SYS 32768, mentre SYS 49152 manderà in esecuzione Monitor $C000.

L'ingresso in "ambiente" monitor è caratterizzato dalla visualizzazione sullo schermo di una serie di sigle e numeri esadecimali, che mostrano il contenuto dei registri della CPU (nel momento in cui è stata attivata la SYS), alcuni dei quali sono già noti: AC sta per Accumulatore, XR e YR indicano i registri X ed Y.

Subito dopo appare un punto (il cosiddetto "prompt") e il cursore lampeggiante: ciò significa che il monitor è in attesa di comandi. Questi, senza alcuna eccezione, sono tutti composti da un singolo carattere, cui vanno aggiunte, dopo uno spazio, le indicazioni (per lo più numeriche) necessarie per operare.

In caso di errore, unica segnalazione sarà la comparsa di un punto interrogativo, dopodichè il cursore andrà a capo, di nuovo in attesa.

Non vale assolutamente la pena lavorare in linguaggio macchina "puro"

Vi sono molti package, tra cui il Macro Assembler Commodore, che rendono il l.m. facile (quasi) come il Basic

Usi ed abusi dei Monitor

L'elaborazione di un programma in linguaggio macchina con il Macro Assembler, passa, nell'ordine, attraverso l'utilizzo dell' Editor, poi dell' Assembler e infine del "loader", che colloca fisicamente in memoria il codice numerico generato dall'Assembler.

Nella sequenza di operazioni appena descritta, i programmi monitor si collocano in una posizione che potremmo definire "ibrida". Per operare correttamente, infatti, il loro uso dovrebbe rappresentare l'ultimo gradino della procedura: quello che consente di ispezionare direttamente le locazioni di memoria in cui è allocato il codice macchina, per verificarne la correttezza o apportarvi minime variazioni, purchè l'utente sia abbastanza esperto per farlo.

Spesso, grazie all'estrema rapidità con cui è possibile assemblare in memoria una sequenza di codici Assembly, si è portati a tentare di realizzare una routine adoperando esclusivamente il Monitor, saltando a piè pari le fasi di editing, assemblaggio e allocazione in memoria, che, al contrario, sono fatti apposta per evitare noie ed errori!

Questo metodo, quindi, è decisamente sconsigliabile, soprattutto se si è alle prime armi con l'Assembly (ma anche in seguito).

Ad un apparente minor numero di "passaggi" da compiere, si accompagna infatti una estrema difficoltà nell'apportare, dopo la stesura, (già di per sè difficoltosa), modifiche anche non sostanziali al programma redatto in tal modo.

Sfruttando, invece, nel modo più corretto tutte le risorse del Macro Assembler (o di altri package similari), si disporrà sempre di routine ben chiare, "trasportabili" e, soprattutto, si progredirà molto più speditamente nella conoscenza del linguaggio Assembly che per molti aspetti risulta piuttosto vicino alla struttura logica, ben nota, del Basic.

L'unico sistema numerico accettato è quello esadecimale, i cui valori devono sempre essere espressi utilizzando 4 cifre (per esempio, 45 dovrà essere digitato 0045); solo in particolari casi (gli operandi in pagina zero) sono accettati valori di due cifre.

La corretta sintassi per ogni comando verrà a mano a mano affrontata nella pratica, ma, giusto come "assaggio", provate a digitare...

```
I A000
```

...dove I sta per "Interrogate" e A000 corrisponde al valore decimale 40960. Vedrete apparire, in reverse, il contenuto ASCII di 8 locazioni di memoria a partire, appunto, da 40960. Se ora vi portate con il cursore in fondo allo schermo e lo fate scrollare verso l'alto, appariranno i contenuti delle locazioni successive; è possibile anche portare il cursore in alto e far scrollare lo schermo verso il basso: come si può dedurre dalla scritta CBMBASIC e (da A09E in poi) dal lungo elenco di parole chiave, ci siamo "introdotti" nell'area di memoria del C/64 adibita all'interpretazione del linguaggio Basic.

Se poi si considera che è possibile modificare a nostra discrezione qualunque locazione o blocco di memoria (purchè in RAM, ovviamente), ecco che ci troviamo di fronte ad uno strumento di estrema potenza, capace di renderci veramente "padroni" di ogni angolo nascosto del nostro computer.

Macroassembler: il file sorgente...

Per riferirsi, concretamente, ad un esempio di facile comprensione, vi invitiamo ad usare il Macro Assembler Commodore per scrivere, in Assembly, la semplice routine proposta nell'articolo "Uno schermo di scorta", pubblicato in questo stesso fascicolo.

Abbiamo visto come, per mettere le ali al C/64, si è dovuto allocare in memoria un gruppo di valori costituenti il linguaggio macchina vero e proprio. Questi, però, rappresentano solo l'ultimo prodotto (quello più incomprensibile) di una più chiara elaborazione che inizia dalla fase di editing. Vediamone nella pratica lo sviluppo.

Anzitutto carichiamo dal dischetto Macro Assembler il programma "editor64" nella forma...

Load "editor64",8,1

...e impartiamo una SYS 49152. Dopo un messaggio di copyright, riapparirà il solito READY ed il cursore lampeggiante, come se nulla fosse accaduto, ma in effetti l'editor è già attivo.

Dei molti nuovi comandi disponibili si è già parlato sul numero scorso della rivista (e se ne riparlerà); per ora limitiamoci ad operare facendo riferimento al disassemblato pubblicato nell'articolo citato.

Anzitutto è necessario digitare un numero di linea, che avrà lo stesso significato che in ambiente Basic, con la differenza che non è possibile farvi riferimento nelle istruzioni Assembly (niente GOTO XX, insomma).

Dopo il numero di linea è obbligatorio uno spazio, e qualunque cosa si trovi dopo questo spazio verrà interpretato dal Macro Assembler come una label (= etichetta) che rappresenta l'unico tipo di riferimento non numerico utilizzabile dai codici mnemonici. Nel disassemblato, potete vedere chiaramente come le label (START, CICLOX, BUF, ecc.) vengano utilizzate.

Per digitare le vere e proprie istruzioni Assembly, bisognerà portarsi almeno due spazi oltre il numero di riga, ma è buona norma mantenere un certo ordine estetico, allineandole tutte al di là di quello che possiamo definire il "campo" delle etichette.

Dopo ancora uno spazio, vanno posti gli operandi delle istruzioni, cui possono seguire eventuali commenti (come le REM del Basic) purchè preceduti da un carattere di punto e virgola (;).

Rispettando queste regole, è possibile trascrivere l'intero disassemblato "trasferimento area video", ricordando di premere il tasto Return al termine di ogni riga, esattamente come se si trattasse di un programma scritto in Basic.

Alla fine della breve digitazione non resta che salvare il lavoro su disco, digitando in modo diretto:

PUT "TRASF"

...o un altro nome di vostro gradimento, e premendo il tasto Return. Se in un secondo tempo si volesse apportare qualche modifica al listato, questo può essere ricaricato nell'Editor usando:

GET "TRASF"

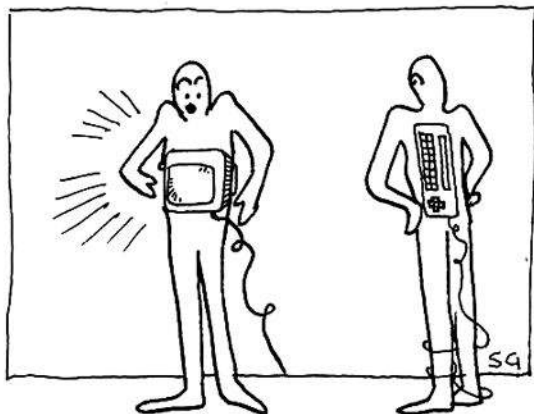
Su disco sarà dunque presente la struttura di base di qualunque elaborazione Assembly: vale a dire il file sorgente, pronto per essere trasformato negli effettivi codici numerici del linguaggio macchina da un altro programma, sempre contenuto nel dischetto Macro Assembler.

...ed il file oggetto

"Usciamo" ora dall'Editor impartendo il comando KILL, o resettando il computer; quindi carichiamo il programma "Assembler64" con la sintassi...

Load "Assembler64",8

La versatilità di un package si valuta anche in base alle opzioni disponibili

SG

A seconda dell'ubicazione della nostra routine l.m. caricheremo il Monitor opportuno

...e lanciamolo con RUN. Rispondiamo alla prima richiesta fornendo il nome che vogliamo assegnare al file che sarà prodotto dall'assembler: ad esempio, TRASF.OBJ. in seguito batteremo Y (Yes) oppure N (No) alla successiva domanda relativa alla copia su carta dell'assemblaggio, opzione molto utile per chi possiede la stampante.

Premiamo semplicemente il tasto Return alla richiesta successiva, quindi, finalmente, forniamo il nome del file da trattare (source file), che è, ovviamente, quello salvato in precedenza: TRASF.

L'assembler si metterà subito al lavoro, creando, sullo stesso dischetto, il file TRASF.OBJ, e segnalando gli eventuali errori commessi nell'editare il programma sorgente. Se ciò è avvenuto, non resta che ricominciare la procedura dall'inizio.

Se, invece, tutto è andato per il verso giusto, disporremo di un file (file oggetto) contenente i veri codici numerici del linguaggio macchina, ma... non è finita qui!

Per utilizzarli in pratica, infatti, è necessario installarli in memoria (come si è fatto in precedenza con il caricatore Basic), adoperando uno dei due programmi "loader" presenti nel dischetto Macro Assembler: "loloader64" e "hiloader64" (scritti proprio così). Il primo, una volta caricato con...

Load "loloader64",8

...si colloca nella normale area Basic (cioè da 2049 in poi) e va quindi lanciato con un normale RUN. Il secondo, da caricare nella forma...

Load "hiloader64",8,1

...va invece ad occupare le locazioni da 51200 in poi e per lanciarlo si deve battere: SYS 51200. Com'è intuibile, si userà uno piuttosto che l'altro, in relazione alla zona di memoria in cui si vogliono allocare i codici del file oggetto.

Nel nostro caso, carichiamo "loloader64" e impartiamo il RUN. Alla richiesta di un offset premiamo semplicemente il tasto Return, poi forniamo il nome del file oggetto da allocare, cioè TRASF.OBJ. Finalmente, il nostro programma LM è in memoria, pronto per essere utilizzato, esplorato e... perchè no, salvato su disco in una forma più comoda da utilizzare in futuro.

Infatti, tanto che si usi il Macro Assembler quanto il caricatore Basic, ogni volta che si intende utilizzare la routine LM occorrerà insediare in memoria il suo codice numerico ricorrendo nel primo caso ai "loader", e nel secondo al programma Basic di caricamento. Ma svincolarsi da queste pastoie è piuttosto semplice, soprattutto se aiutati da un'altra "feature" del Macro Assembler, cioè i suoi Monitor. Vediamo come.

Siamo giunti, con le operazioni prima illustrate (o semplicemente caricando e attivando il programma n.2), ad avere in memoria la nostra routine LM, che occupa le locazioni di memoria da 50176 a 50217. Senza spegnere il computer, diamo semplicemente un NEW o, se ne siete in possesso, premete il pulsantino di reset, vivamente consigliato per lavorare più speditamente in LM, quindi digitate...

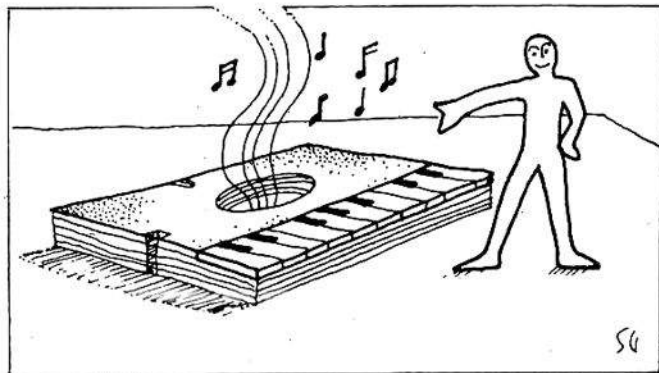
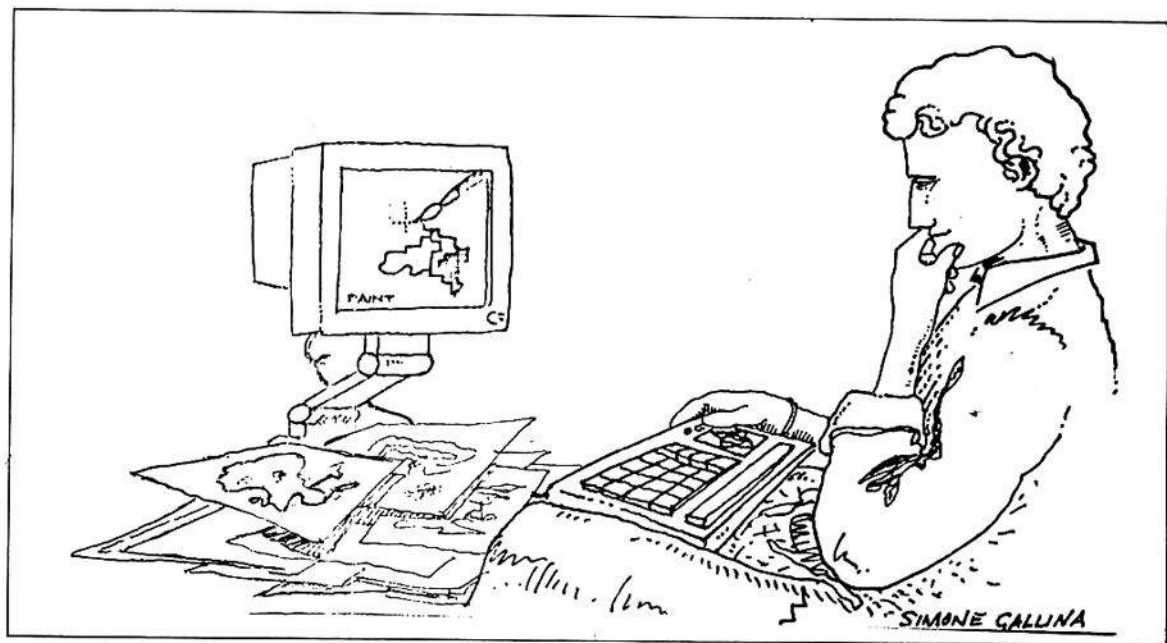
Load "monitor$8000",8,1

...e premete Return. Ultimato il caricamento, impartite l'istruzione SYS 32768 e, una volta entrati in ambiente monitor, digitate esattamente...

S "TRASF.LM",08,C400,C42A

... che salverà (S = Save) con il nome TRASF.LM, in un file su disco (08), il contenuto delle locazioni di memoria da 50176 (in esadecimale: C400) a 50218 (esadec. = C42A).

Ogni qual volta ne avrete bisogno, basterà caricare direttamente il file TRASF.LM (utilizzando la forma ....,8,1) per avere subito in memoria la nostra routine, indipendentemente dalle routine di aiuto contenute nel disco Macro Assembler.

SIMONE GALUNA

UN DETECTIVE GRAFICO

Un breve programma si rende utile per visualizzare le varie pagine grafiche, ed i font di caratteri, allocati nella memoria del C/64

di **Bruno Cattivelli**

Il listato di queste pagine è scritto in Basic, ma utilizza una piccola routine in l.m. situata nelle linee Data 115 e 120 il cui disassemblato è pubblicato a parte.

Il commento del programma, linea per linea, è certamente alla portata di tutti:

• linea 130: vengono inizializzate le variabili "B" e "C" che servono alla gestione della Bit-Map; viene poi settato in nero il colore dello sfondo ed inserita la funzione repeat su tutti i tasti.

• linea 140: a partire da 2816 viene locata la subroutine in l.m. contenuta nelle linee 320-330; l'opportuno comando di linea 120 (Poke 56,11) provvede a fissare il Top di memoria in modo da proteggere la stessa routine.

• linea 160: lo schermo viene ripulito e viene controllato il valore di M che deve essere compreso tra 0 e 65535.

• linea 170: calcola, in base ad M, i valori da inserire nelle locazioni 251-252 nella forma Low-byte / High-byte.

• linea 180: poka il valore di M (Lb / Hb) nelle locazioni libere di pagina zero 251-252.

• linea 190: chiama la subroutine di copiatura in l.m. e setta la pagina grafica (Bit-Map).

• linea 200: attende la pressione di un tasto.

• linea 210: se il tasto è Return allora restituisce sullo schermo i valori iniziale e finale della porzione di 8k visualizzata nella Bit-Map.

• linee 220-290: avanza (o arretra) nella memoria incrementando o decrementando il valore di M a seconda del tasto premuto. Poi ricomincia.

• linea 300: ignora la pressione di tasti errati.

Localizzare una pagina grafica è un'operazione molto utile

Una manciata di righe è sufficiente per esplorare la memoria del vostro C/64

• linee 320-330: routine in l.m. che copia una porzione di 8000 byte (il cui inizio è situato nelle locazioni 251-252) nella pagina grafica 8192-16192.

COME FARLO GIRARE

Il programma è semplice da usare perchè visualizza diverse porzioni di pagina grafica a seconda del tasto funzione premuto:

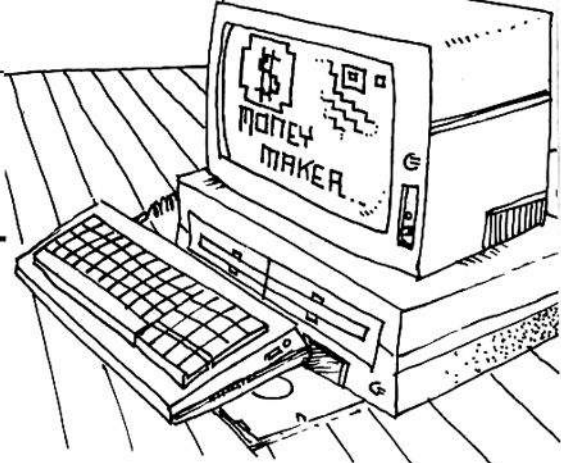
- F1: un byte avanti
- F2: un byte indietro
- F3: un carattere avanti (8 byte)
- F4: un carattere indietro
- F5: una riga avanti (320 byte)
- F6: una riga indietro
- F7: un kbyte avanti (1024)
- F8: un kbyte indietro

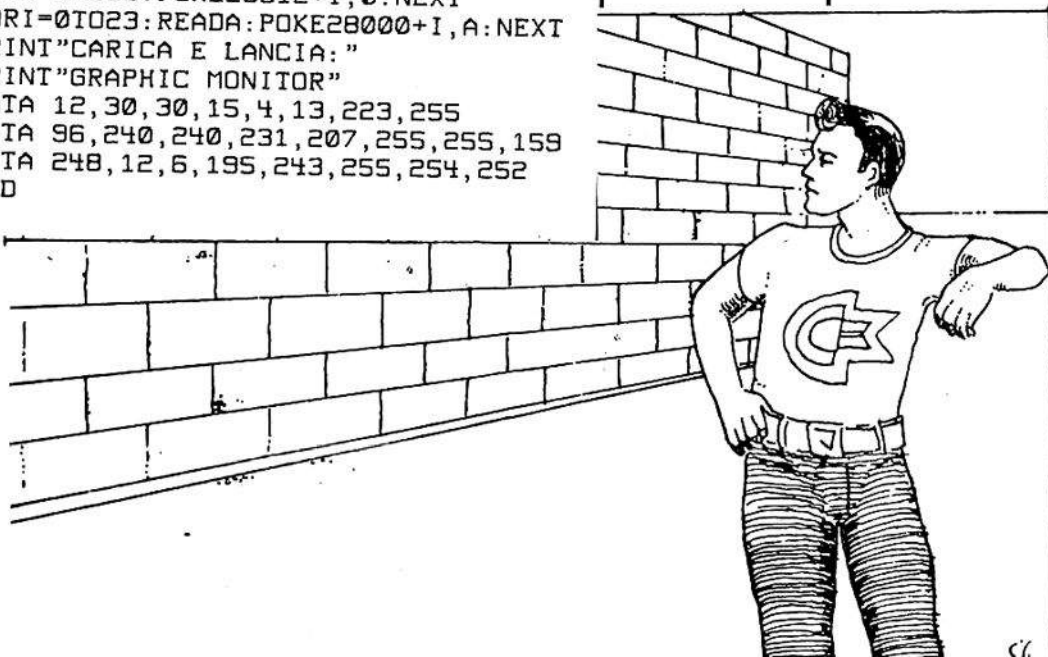
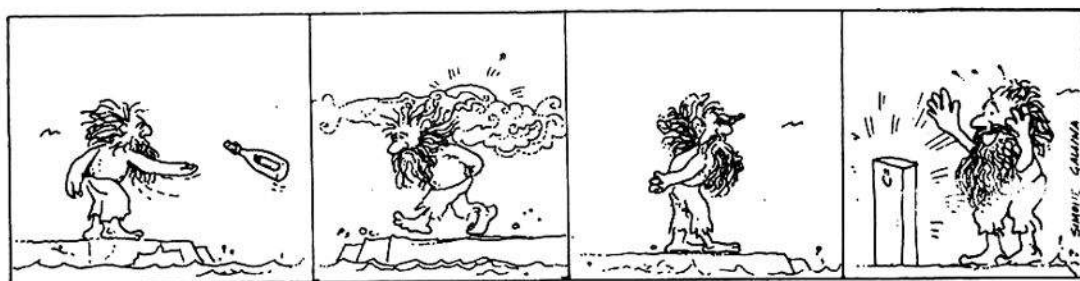
Premendo il tasto Return scompare la bit map e compare il valore attuale di M e di M+8192. Premendo un tasto funzione la visualizzazione riprende.

Il programmino dimostrativo, pubblicato a parte, disegna un topolino in una zona della memoria: tocca a voi scoprirlo (e spostarlo) agendo sui tasti funzione.

Concludiamo ricordando che eventuali pagine grafiche multicolor verranno riprodotte in modo pressochè illeggibile a causa del diverso formato della pagina stessa: ai più bravi il compito di introdurre una modifica idonea per visualizzare anche le pagine multicolor, come pure la possibilità di rilocare la routine in altra zona di memoria.

Ricordiamo, inoltre, di non renumerare il programma a causa del comando Goto 150 contenuto in riga 310 che non verrebbe renumerato; non dimenticare, inoltre, che a causa dell'ubicazione della stessa routine l.m. la memoria disponibile è di una trentina di byte: tenetene conto nel caso compaia lo sgradito messaggio "Out of memory error".

```
100 REM GRAPHIC MONITOR
110 REM BY B.CATTIVELLI
120 POKE56,11:RUN130
130 B-53272:C-53265:POKE53280,0:POKE650,128
140 FORI-0TO29:READA:POKE2816+I,A:NEXT
150 M-8192
160 PRINT"]":IFM<0ORM>65536THEN310
170 HB-INT(M/256):LB-M-HB*256
180 POKE251,LB:POKE252,HB
190 SYS2816:POKEB,29:POKEC,59
200 GETA$:IFA$-""THEN200
210 IFA$-CHR$(13)THENPOKEB,21:POKEC,27:PRINTM;M+8192:PRINT"■■■CONT]]]":END
220 IFA$-CHR$(133)THENM-M+1:GOTO160
230 IFA$-CHR$(137)THENM-M-1:GOTO160
240 IFA$-CHR$(134)THENM-M+8:GOTO160
250 IFA$-CHR$(138)THENM-M-8:GOTO160
260 IFA$-CHR$(135)THENM-M+320:GOTO160
270 IFA$-CHR$(139)THENM-M-320:GOTO160
280 IFA$-CHR$(136)THENM-M+1024:GOTO160
290 IFA$-CHR$(140)THENM-M-1024:GOTO160
300 GOTO200
310 POKEB,21:POKEC,27:PRINT"■■■GOTO150]]]":END
320 DATA 162,0,160,0,169,32,133,254,138,133,253,177,251,145,253,200,208,249,160
330 DATA 0,230,252,230,254,232,224,33,208,238,96
```

```
100 REM DEMO GRAPHIC MONITOR
110 :
120 REM PROVA A TROVARE LA TANA DEL
130 REM TOPO NASCOSTO NELLA MEMORIA
140 :
150 FORI=0TO39:POKE27672+I,0:NEXT
160 FORI=0TO39:POKE27992+I,0:NEXT
170 FORI=0TO39:POKE28312+I,0:NEXT
180 FORI=0TO23:READA:POKE28000+I,A:NEXT
190 PRINT"CARICA E LANCIA:"
200 PRINT"GRAPHIC MONITOR"
210 DATA 12,30,30,15,4,13,223,255
220 DATA 96,240,240,231,207,255,255,159
230 DATA 248,12,6,195,243,255,254,252
240 END
```

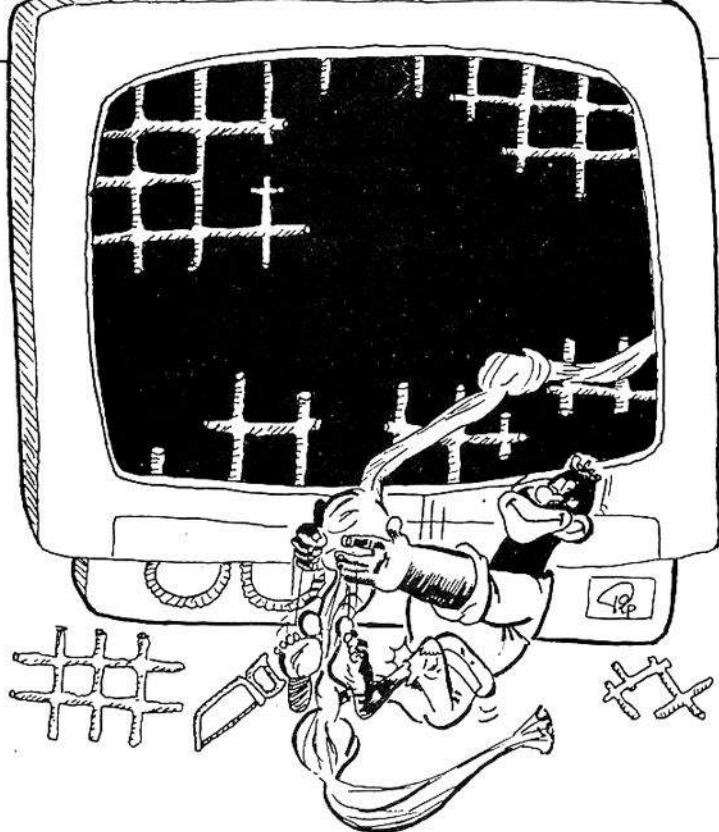
Digitate correttamente il semplice listato, soprattutto le linee Data

Se non conoscete il l.m. potete egualmente utilizzare il programma proposto

```
Disassemblato della routine LM
      GRAPHIC MONITOR


0D00 LDX #$00    :azzera registro X.
     LDY #$00    :azzera registro Y.
     LDA #$20    :carica l'accumulatore
                  con l'indirizzo 8192
                  (Hi-byte di $2000).
     STA $FE     :e lo memorizza nella
                  locazione 254 (libera
                  pagina zero).
     TXA         :azzera l'accumulatore.
     STA $FD     :carica nella loc.253
                  il Low-byte di $2000.
0D0B LDA ($FB),Y:copiatura. In $FB-$FC
                  c'e' l'inizio della
                  porzione da copiare.
     STA ($FD),Y:copiatura. In $FD-$FE
                  c'e' la porzione in
                  cui copiare.
     INY         :incrementa Y di uno.
     BNE $0D0B   :ricomincia finche' Y
                  non torna a zero.
     LDY #$00    :azzera Y.
     INC $FC     :incrementa l'Hi-byte
                  della loc. da copiare.
     INC $FE     :incrementa l'Hi-byte
                  della locazione in
                  cui copiare.
     INX         :incrementa X che serve
                  da contatore.
     CPX #$21    :controlla se il conta-
                  tore ha finito
                  (32*256=8192).
     BNE $0D0B   :se non ha finito
                  continua a copiare.
     RTS         :se ha finito ritorna
                  al basic.
```

AL DI LA' DELLE SBARRE DEL VIDEO

Pubblichiamo, qui di seguito, il disassemblato relativo all'articolo dello scorso numero che non e' apparso per un errore della redazione. Ci scusiamo con i lettori per il disguido.

```
 1 **** LISTATO SOLO PER C-64 ****
 2 ********************************
 3 * QUESTO PROG. , GESTITO DA UN *
 4 * IRQ, APRE UNA FINESTRA NEL   *
 5 * LATO SUPERIORE E POSTERIORE  *
 6 * DEL BORDO DI SCHERMO.        *
 7 * IN QUESTA PORZIONE POSSONO   *
 8 * ESSERE VISUALIZZATI SOLO     *
 9 * DEGLI SPRITE.                *
10 ********************************
11 *        SIDE BORDER 1         *
12 *     BY DAVIDE ARDIZZONE      *
13 *   DISASSEMBLATO PER C-64.    *
14 ********************************
15 *       INIZIO PROGRAMMA
16 ********************************
17 RASTER   =     $D012        ;REGISTRO DEL RASTER.
18 ********************************
19 START     SEI               ;SETTA L'IRQ A UNO.
20          LDA   #<IRQ        ;BYTE BASSO.
21          LDX   #>IRQ        ;BYTE ALTO.
22          STA   $0314        ;QUESTI DUE INDIRIZZI
23          STX   $0315        ;INDICANO LA POSIZIONE
24                             ;DEL PROGRAMMA IRQ
25                             ;NELLA MEMORIA.
26          LDA   #$80         ;IL PENNELLO OTTICO
27          STA   RASTER       ;ASSUME IL VALORE 128.
28          LDA   $D011        ;AZZERA
29          AND   #$7F         ;IL COMPARATORE
30          STA   $D011        ;DI QUADRO.
31          LDA   #$F0         ;DISABILITA ESECUZIONI
32          STA   $D01A        ;NON NECESSARIE AL
33                             ;NOSTRO IRQ.
34          LDA   #$01         ;ATTIVA TIMER DEL
35          STA   $DC0E        ;REGISTRO 'A' DEL CIA #1.
36          LDA   #$02         ;COLORE ROSSO
37          STA   $D020        ;NEL BORDO.
38          LDA   #$00         ;COLORE NERO
39          STA   $D021        ;NELLA PAGINA.
40          CLI                ;ATTIVA IRQ.
41          RTS                ;RITORNO AL BASIC.
```

```
42 ********************************
43 *INIZIO SCANSIONE DELL'IMMAGINE
44 ********************************
45 IRQ       LDA   #$01      ;ATTIVA IL CONTROLLO
46           STA   $D019     ;DI CONTATTO TRA
47                           ;SPRITE E BORDO.
48           LDA   #$F8      ;ULTIMO VALORE NUMERICO
49                           ;DELL'ORDINATA DELLO
50                           ;SCHERMO.
51 COMP      CMP   RASTER    ;COMPARA CON LA
52                           ;POSIZIONE ATTUALE DEL
53                           ;PENNELLO OTTICO.
54           BNE   COMP      ;SALTA SE E'DIFFERENTE.
55           LDA   #$13      ;VALORE NUMERICO DA
56           STA   $D011     ;ATTRIBUIRE PER
57                           ;L'APERTURA DEL BORDO.
58           LDA   #$20      ;PRIMO VALORE NUMERICO
59                           ;DELLA PAG. DI SCHERMO.
60 COMP1     CMP   RASTER    ;COMPARA CON IL PENNELLO
61                           ;OTTICO (RASTER).
62           BNE   COMP1     ;SALTA SE E' DIVERSO.
63           LDA   #$1B      ;VALORE STANDARD DELL'
64                           ;EDITOR DI SCHERMO.
65           STA   $D011     ;REGISTRO DI CONTROLLO
66                           ;DELLO SCHERMO (6566).
67           JMP   $EA31     ;SALTA ALLE PROCEDURE
68 ********** FINE **********;DEL SISTEMA.
```

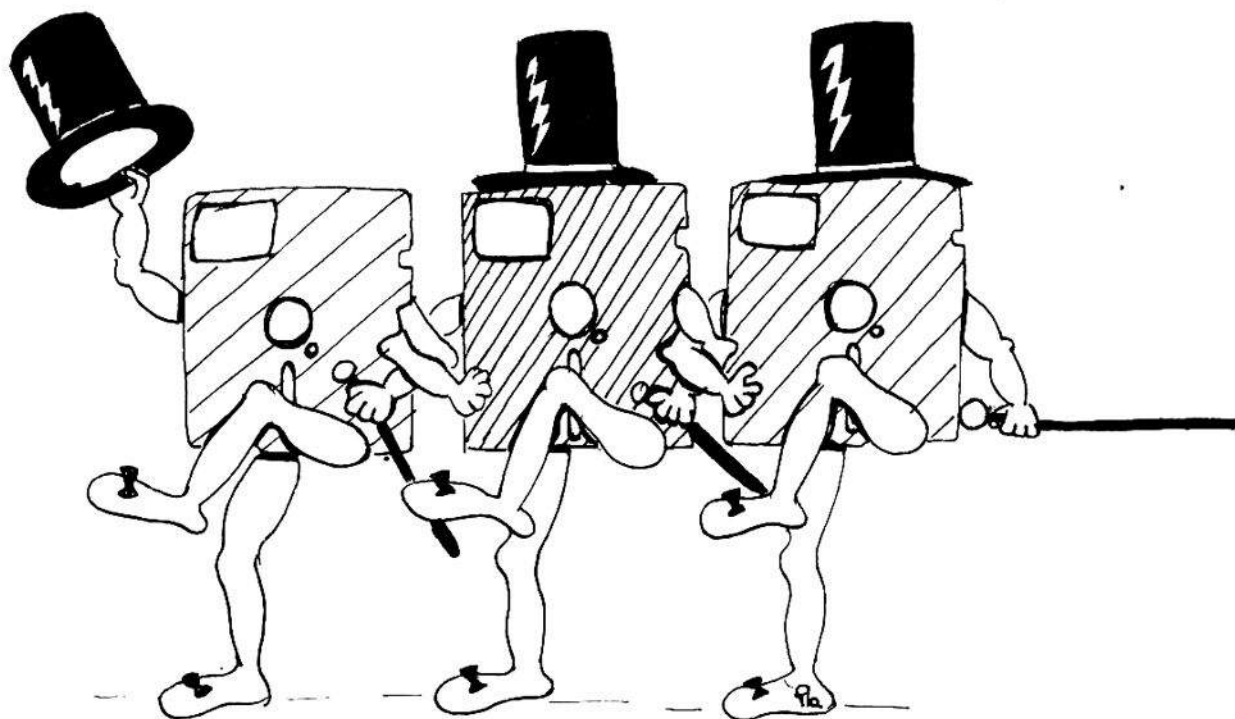
IN EDICOLA

MS-DOS

&

GW-BASIC

EMULATOR 2.0

systems

PASSEGGIANDO SULLO SCHERMO

La realizzazione di scritte scorrevoli sul video è solo una scusa per avventurarsi all'interno del TED Chip: il nostro scopo è scoprire le più nascoste potenzialità del C/16 e del Plus/4

di **Roberto Ferro**

Molti di voi sapranno che nel C/64 la gestione della grafica e del suono è affidata a due chip interni specializzati: il VIC-II (per la grafica) ed il SID (per il suono). I sessantaquattristi possono ottenere gli effetti di maggiore interesse solo ricorrendo alla manipolazione dei registri di questi chip. Un discorso analogo può essere fatto per il C/16 ed il Plus/4.

Con questi computer, generalmente, non è il caso di ricorrere al linguaggio macchina per sfruttare caratteristiche come il suono o la grafica perchè esiste un comodo e completo set di comandi Basic che facilita la vita. In realtà, però, esistono alcune operazioni che neanche il Basic V3.5 riesce a gestire, come ad esempio la manipolazione del raster, la gestione del video a 38 colonne o, ancora, lo scrolling pixel per pixel: tutte cose che vedremo di realizzare in questo articolo.

Se pensate che si tratti di argomenti troppo complessi da capire (ma è una spudorata bugia!) potete limitarvi a digitare il listato; se, invece, avete buona volontà, pazienza e voglia di imparare, allora state tranquilli e seguiteci, perchè ci avventureremo in un mondo affascinante, pieno di piacevoli imprevisti e pro-

SCHEDA TECNICA

Software di utilità per applicazioni grafiche.

Hardware richiesto: C/16, Plus/4; non adattabile ad altri computer Commodore

Consigliato ai principianti del linguaggio macchina

Utile anche agli inesperti, purchè digitino con la massima attenzione il programma pubblicato.

Anche il programma pubblicato in queste pagine è contenuto nel disco "Directory" di questo mese.

babilmente meno ostico di quanto avreste osato pensare.

COMINCIA L'AVVENTURA

Il circuito integrato interno del C/16 e Plus/4 che si occupa di grafica è lo stesso che controlla il suono e risponde al nome di TED Chip. E' chiaro che, rispetto al VIC-II o al SID del C/64, il piccolo TED offre potenzialità minori e lo capiamo anche dal fatto che non si tratta di un chip specializzato solo per la grafica o solo per il suono, ma di un chip che cerca di fare il possibile sia nell'uno che nell'altro campo.

Il TED ha dei registri accessibili dal momento che sono contenuti in alcune locazioni di memoria, e più precisamente nell'intervallo $FF00-$FF3F. Questo significa che volendo agire sul registro 21 che controlla il colore dello sfondo e che è contenuto in $FF15 potremo impartire una semplice POKE nella locazione 65301. Altri registri sono già stati esaminati in precedenti articoli, come ad esempio il 18 ($FF12) ed il 19 ($FF13) per la ridefinizione dei caratteri.

I registri di cui andiamo ad occuparci ora sono il 7, il 9, l'11 e il 29.

IL MOVIMENTO PIXEL PER PIXEL

Il registro 7 è contenuto nella locazione $FF07 e, come tutti i byte, è composto da 8 bit. I bit che interessano sono i primi 4 (numerati da 0 a 3) mentre per i rimanenti 4 bit diciamo solo che è necessario che siano posti sempre a 0.

La struttura del registro 7 è mostrata in figura 1. I primi tre bit si occupano dello scrolling orizzontale del video, mentre il quarto seleziona il modo di visualizzazione a 40 oppure 38 colonne; ma andiamo con ordine. Provando ad impartire...

Print Peek(65287)

...otterremo in risposta il valore 8. Come è noto dalla notazione binaria, il decimale 8 indica che solo il quarto bit, cioè il n.3, è posto ad 1, mentre tutti gli altri sono a zero. Sappiamo quindi che in condizioni normali i bit che governano lo scrolling

RASTER REGISTER E DINTORNI

Riguardo a concetti come raster, pennello elettronico e linee di scansione, si è già discusso sul numero 47 di C.C.C, seppur con esplicito riferimento al C/64.

Nonostante la gestione del video non sia molto differente in un C/16 o Plus/4, diamo ora un breve cenno sia sul meccanismo che il computer utilizza per visualizzare un'immagine, che sulle differenze rispetto a un C/64.

Quando il monitor deve visualizzare un'immagine trasmessa dal computer, considera lo schermo suddiviso in sottilissime linee orizzontali, 312 per la precisione. All'interno del monitor esiste un particolare circuito che emette un sottile fascio di elettroni sullo schermo. Questa emissione avviene tenendo conto della suddivisione dello schermo in linee: il fascio di elettroni (comunemente chiamato cannone, o pennello elettronico) percorre, una per una, tutte le linee, spegnendosi e riaccendendosi ogniqualvolta giunge al termine di una di esse e ne ricomincia un'altra sempre partendo da sinistra.

Gli elettroni accendono o lasciano spenti i fosfori del video provocando la comparsa dell'immagine.

All'interno del TED Chip esistono due registri che si occupano di controllare il pennello elettronico: il 29 e l'11. Il primo è contenuto in $FF1D e riporta il "corrente" numero di linea scandita. Il secondo è contenuto in $FF0B e vedremo tra poco la sua funzione.

Noi, però, sappiamo che le linee di scansione sono più di 255 ed infatti il registro 29, che ne riporta il numero, è in realtà composto da nove bit: otto sono contenuti in 29; il nono è il bit 0 del registro 28. Il registro 29 può solamente essere letto e, come intuibile, il suo valore cambia molto rapidamente.

Il registro 11, invece, contiene quello che viene definito il valore per la comparazione della scansione dell'immagine.

Il valore contenuto in questa locazione funge da test per quello contenuto in $FF1D: quando $FF1D (registro 29) raggiunge il valore contenuto in $FF0B, viene generato un interrupt di raster che, opportunamente intercettato, può essere usato per visualizzare uno schermo suddiviso in più zone, ognuna delle quali con proprie caratteristiche.

Per controllare se l'interrupt corrente è di raster bisogna testare il bit 2 del registro 9 ($FF09) e controllare che non si tratti di un interrupt generato da un altro tipo di timer.

Questo controllo è la prima operazione che compie la routine LM ScrollPix.

Analizzando attentamente il disassemblato di ScrollPix si può capire che con un'attenta manipolzione del raster register si possono ottenere effetti veramente eccezionali, altrimenti impossibili da realizzare.

orizzontale sono azzerati, mentre il bit di controllo del modo a 40 o 38 colonne è a 1.

Possiamo cominciare a fare qualche prova alterando il contenuto di questo registro. I bit per lo scrolling possono rappresentare un numero tra 0 e 7 e questo numero indica di quanti pixel l'area del testo sarà spostata verso destra. Se quindi vogliamo uno spostamento di un pixel dobbiamo immettere il valore 1, se lo vogliamo di due pixel il valore 2, se di tre il 3 e così via. Dobbiamo però ricordarci, almeno per ora, di non alterare il contenuto del quarto bit. Praticamente tutto ciò si traduce nel fatto che i valori validi vanno aumentati di 8 e sono quindi quelli compresi tra 8 e 15. Il seguente programmino offre un esempio degli effetti che è possibile ottenere agendo sui bit per il controllo dello scrolling:

```
10 poke 65287,8
20 a=peek(65287)
30 if (a and 7)=7 then end: else
a=a+1
40 for i=0 to 100: next
50 poke 65287,a: goto 20
```

Lanciatolo, vedremo la zona del testo muoversi pixel per pixel verso destra.

Purtroppo esistono due problemi. Innanzitutto l'immagine, spostatasi verso destra, lascia sulla sinistra una brutta colonna di caratteri bianchi; in secondo luogo abbiamo spostato lo schermo di soli 8 pixel mentre è auspicabile che lo spostamento avvenga per un qualsiasi numero di pixel.

La soluzione al primo problema la offre lo stesso TED. Come forse alcuni avranno già capito, ci serviremo del bit n.3 del registro 7. Grazie ad esso decidiamo se visualizzare un'area testo a 38 oppure a 40 colonne. Se infatti lo poniamo a zero, abilitando il modo 38 colonne, abbiamo risolto il problema della colonna di caratteri bianchi che viene ad essere coperta dal bordo. Per verificare l'effettiva utilità di questo bit, impartiamo...

Poke 65287,0

...oppure inseriamo la linea...

5 Poke 65287,0

...all'inizio del programmino visto prima.

Rimane, però, ancora senza soluzione il problema di spostare una immagine per più di 8 pixel. In questo caso il TED non ci viene molto in aiuto e così dovremo arrangiarci per conto nostro.

Sappiamo che il numero massimo di spostamenti in pixel effettuabili ricorrendo al registro locato in $FF07 è sette. Possiamo però renderci conto che l'ottavo spostamento può essere simulato con lo spostamento dell'intera colonna di caratteri una locazione in avanti.

In parole povere possiamo dire che spostare una colonna di caratteri di otto pixel equivale a spostarla di un carattere e quindi il discorso va posto in questi termini: fino al settimo pixel ci serviamo dello scrolling hardware mentre, per l'ottavo, poniamo a zero il registro 7 e spostiamo l'intera pagina testo di un carattere. Anche se può sembrare un po' contorto, il meccanismo è molto logico e neanche troppo complesso. Un ultimo particolare va però tenuto presente: se non siamo abbastanza veloci ad azzerare il registro ed a spostare la pagina testo avremo un effetto di "inceppamento", con uno scrolling che, ogni otto pixel, subisce una battuta d'arresto.

E' chiaro che per quest'ultimo problema la soluzione è offerta solo dal linguaggio macchina. Il listato di queste pagine è infatti una routine LM che gestisce le operazioni suddette.

SCROLLPIX

La routine è molto utile se usata nelle schermate di presentazione. Essa infatti si "impadronisce" dell'ultima linea di testo e la gestisce facendone scrollare il contenuto verso sinistra.

Prima di proseguire è necessaria una pecisazione: quando spostiamo una linea di testo verso destra, o verso sinistra, il primo carattere viene a scomparire, mentre l'ultimo risulta vuoto. A questo punto possiamo effettuare due scelte: visualizzare da un lato ciò che esce da un altro, oppure immettere caratteri nuovi e lasciare perdere i primi. La soluzione adottata dal programma di queste pagine è la prima, ed è quella che offre una visione "rotatoria" del messaggio che, però, è limitato alla lunghezza della linea stessa.

Per utilizzare ScrollPix dobbiamo stampare il messaggio desiderato nell'ultima linea e poi richiamare la

Registro 7 del TED Chip ($FF07)

Reverse Spento	PAL/NTSC	Congelatore	Multicolor	38/40 colonne	Scroll Orizz. 2	Scroll Orizz. 1	Scroll Orizz. 0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

routine con SYS 16224. La stessa routine provvederà, subito dopo, a creare una finestra in modo che non si possa più accedere all'ultima linea.

Per disattivarla dobbiamo dare la SYS indicata dallo stesso programma, che è auto dimostrativo e auto esplicativo.

Sembrerebbe tutto a posto: la routine l'abbiamo descritta, il movimento pixel pixel anche, ma... come fa ScrollPix a girare contemporaneamente al programma Basic e ad intervenire solo sull'ultima linea di testo? Le risposte sono: Interrupt e Raster register.

E' la prima volta che ci occupiamo di questi argomenti con riferimento specifico al C/16 e Plus/4; sarà quindi bene dare un'occhiata al riquadro, nonchè (se li possedete) ai numeri 31, 44, 47 e 50 di C.C.C.

In linea di massima l'interrupt presenta meno problemi; se avete letto, magari fugacemente, i numerosi articoli su questo argomento, dovreste già avere un'idea piuttosto precisa al riguardo.

Il raster risulta essere decisamete più complesso, ma non scoraggiamoci.

Come spiegato nel riquadro, i registri 11 e 29 ($FF0B e $FF1D) contengono, rispettivamente, il valore per la comparazione della linea di scansione dell'immagine e la linea di scansione stessa. Scrivendo dunque un valore nel registro 11 otteniamo la generazione di un interrupt quando la linea di scansione è giunta al valore immesso. Intercettata questa condizione, possiamo alterare le caratteristiche del display e dividerlo in due "regioni", ognuna con caratteristiche diverse. Potremmo, ad esempio, visualizzare fino a metà video uno sfondo colorato di verde; da quel punto in poi, invece, attivare uno sfondo colorato di blu. Facendo girare una routine di questo tipo, in interrupt, avremo uno schermo verde e blu.

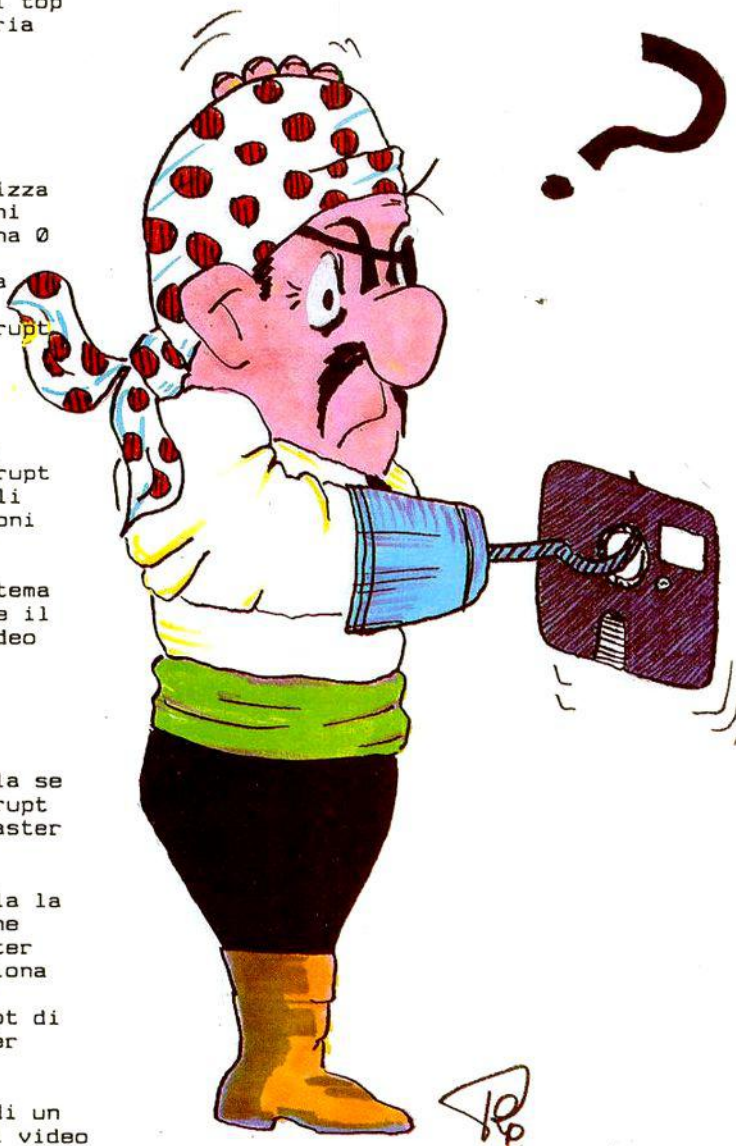
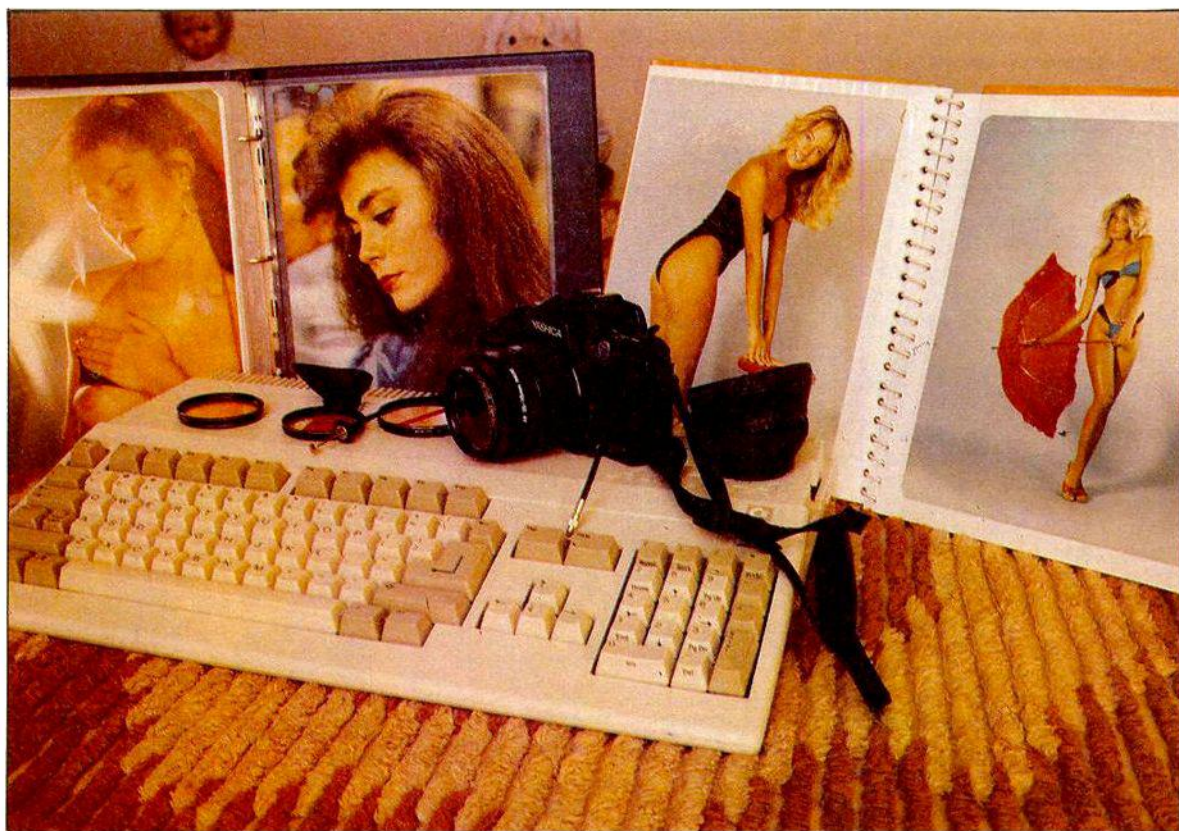
Per ScrollPix il discorso è analogo: per le prime ventiquattro linee gestisce normalmente il display (e tiene presente che bisogna generare un interrupt di raster prima della venticinquesima); quando l'interrupt della venticinquesima linea viene generato (corrisponde alla linea di scansione 194 o $C2), allora fa in modo che il display, da quel punto in poi, scorra pixel per pixel.

Ulteriori spiegazioni sono riportate nel disassemblato commentato.

```
10 REM **************************
20 REM       SCROLLPIX  V3.0
25 REM  SOLO PER C/16 & PLUS/4
30 REM **************************
40 :
50 REM BY ROBERTO E PIETRO FERRO
60 :
70 POKE51,80:POKE52,63
75 PRINTCHR$(147)+CHR$(147)
80 COLOR0,7,2:COLOR4,7,2:COLOR1,2
90 FOR I=16224 TO 16365
100 READ A$:POKEI,DEC(A$)
110 CK=CK+DEC(A$)
120 NEXT I
130 IF CK<>16552 THEN 220
140 SCNCLR
150 PRINTCHR$(17)+" SCROLLPIX V3.0:"
160 PRINTCHR$(17)+" SYS 16224 - ABILITA
170 PRINTCHR$(17)+" SYS 16262 - DISABILIT
180 CHAR 1,2,24,"COMMODORE COMPUTER CLUB"
190 SYS 16224:REM ATTIVA LA ROUTINE
200 POKE205,10
210 END
220 SCNCLR:PRINT"ERRORE NEI DATA !!!"
230 END
240 :
1000 DATA A9,50,85,33,85,35,85,37
1010 DATA A9,3F,85,34,85,36,85,38
1020 DATA A9,08,85,D8,A9,17,8D,E5
1030 DATA 07,78,A9,A0,8D,14,03,A9
1040 DATA 3F,8D,15,03,58,60,78,A9
1050 DATA 0E,8D,14,03,A9,CE,8D,15
1060 DATA 03,58,A9,08,8D,07,FF,A9
1070 DATA 18,8D,E5,07,60,00,00,00
1080 DATA AD,09,FF,29,02,D0,03,4C
1090 DATA BE,FC,8D,09,FF,AD,1D,FF
1100 DATA C9,C2,B0,10,A9,08,A2,C2
1110 DATA 8D,07,FF,8E,0B,FF,20,9F
1120 DATA FF,4C,BE,FC,A6,D8,CA,8E
1130 DATA 07,FF,D0,05,20,D8,3F,A2
1140 DATA 08,86,D8,A2,00,4C,BB,3F
1150 DATA AD,C0,0F,48,A0,00,B9,C1
1160 DATA 0F,99,C0,0F,C8,C0,27,D0
1170 DATA F5,68,8D,E7,0F,60
```

```
*------------------------------------*
* ROUTINE SCROLLPIX per C/16 & Plus/4 *
* by Roberto Ferro                    *
*------------------------------------*

3f60  a9 50     lda #$50  ;fissa il top
3f62  85 33     sta $33   ;di memoria
3f64  85 35     sta $35
3f66  85 37     sta $37
3f68  a9 3f     lda #$3f
3f6a  85 34     sta $34
3f6c  85 36     sta $36
3f6e  85 38     sta $38
3f70  a9 08     lda #$08  ;inizializza
3f72  85 d8     sta $d8   ;locazioni
3f74  a9 17     lda #$17  ;in pagina 0
3f76  8d e5 07  sta $07e5;e crea
3f79  78        sei       ;finestra
3f7a  a9 a0     lda #$a0  ;devia
3f7c  8d 14 03  sta $0314;l'interrupt
3f7f  a9 3f     lda #$3f
3f81  8d 15 03  sta $0315
3f84  58        cli
3f85  60        rts
3f86  78        sei       ;riporta
3f87  a9 0e     lda #$0e  ;l'interrupt
3f89  8d 14 03  sta $0314;a normali
3f8c  a9 ce     lda #$ce  ;condizioni
3f8e  8d 15 03  sta $0315
3f91  58        cli
3f92  a9 08     lda #$08  ;e risistema
3f94  8d 07 ff  sta $ff07;il TED e il
3f97  a9 18     lda #$18  ;e il video
3f99  8d e5 07  sta $07e5
3f9c  60        rts
3f9d  00        brk
3f9e  00        brk
3f9f  00        brk
3fa0  ad 09 ff  lda $ff09;controlla se
3fa3  29 02     and #$02  ;l'interrupt
3fa5  d0 03     bne $3faa;e' di raster
3fa7  4c be fc  jmp $fcbe
3faa  8d 09 ff  sta $ff09
3fad  ad 1d ff  lda $ff1d;controlla la
3fb0  c9 c2     cmp #$c2  ;posizione
3fb2  b0 10     bcs $3fc4;del raster
3fb4  a9 08     lda #$08  ;riposiziona
3fb6  a2 c2     ldx #$c2  ;video e
3fb8  8d 07 ff  sta $ff07;interrupt di
3fbb  8e 0b ff  stx $ff0b;di raster
3fbe  20 9f ff  jsr $ff9f
3fc1  4c be fc  jmp $fcbe
3fc4  a6 d8     ldx $d8   ;sposta di un
3fc6  ca        dex       ;pixel il video
3fc7  8e 07 ff  stx $ff07
3fca  d0 05     bne $3fd1

3fcc  20 d8 3f  jsr $3fd8
3fcf  a2 08     ldx #$08  ;ripristina la
3fd1  86 d8     stx $d8   ;locazione $d8
3fd3  a2 00     ldx #$00
3fd5  4c bb 3f  jmp $3fbb
3fd8  ad c0 0f  lda $0fc0;sposta la
3fdb  48        pha       ;scritta di un
3fdc  a0 00     ldy #$00  ;carattere
3fde  b9 c1 0f  lda $0fc1,y
3fe1  99 c0 0f  sta $0fc0,y
3fe4  c8        iny
3fe5  c0 27     cpy #$27
3fe7  d0 f5     bne $3fde
3fe9  68        pla
3fea  8d e7 0f  sta $0fe7
3fed  60        rts
```

GLI INTERROGATIVI DEL C

Le strutture di controllo del C sono simili a quelle di molti altri linguaggi strutturati.

di **Luigi Callegari**

Le istruzioni di controllo del flusso di un programma sono sostanzialmente quelle che specificano in che ordine eseguire le operazioni, in base a parametri variabili. Prima di descrivere in dettaglio le istruzioni condizionali e di controllo vogliamo specificare il significato delle seguenti due linee in C:

gabry = 27 + y
gabry = 27 + y;

La prima è una semplice espressione; la seconda è invece una istruzione. Infatti il punto e virgola (;) in linguaggio C indica (come il doppio punto in Basic) la fine di una espressione compiuta e dunque un'istruzione. Più istruzioni possono essere racchiuse tra parentesi graffe per indicare, praticamente, una sola grande istruzione detta, in termini appropriati, un "blocco" o "istruzione composta".

Detto ciò, vediamo le strutture di controllo partendo dalla più familiare: If - Else. La sua sintassi formale è indicata in figura 1.

Se il "blocco istruzioni" è una singola istruzione, le parentesi graffe non sono ovviamente necessarie. La parte comprendente Else è opzionale.

La condizione può essere un'espressione qualunque: se il suo valore è diverso da zero (cioè, vero) viene eseguito il primo blocco, altrimenti (else, in inglese), viene eseguito il secondo blocco di istruzioni, se specificate.

Vediamo, come esempio, il caso in cui se la variabile di carattere "c"

contiene "N" viene terminato il programma con la visualizzazione di un messaggio, altrimenti viene eseguita una determinata funzione (elabora) come indicato in figura 2.

Si noti l'uso dell'operatore di confronto costituito da due segni di eguale (=) affiancati. Se si fosse usato un solo uguale (operatore di assegnazione) la prima linea avrebbe assegnato alla variabile "c" il valore "n" e, essendo tale valore diverso da zero, avrebbe terminato l'esecuzione indipendentemente dal suo contenuto.

Abbiamo detto che la parte Else è opzionale, perciò spesso si fanno delle confusioni quando si aggregano più istruzioni IF-Else una dentro l'altra (nidificazione). In questi casi si deve ricordare che un Else si riferisce sempre allo If più interno, ovvero al più vicino If privo di Else.

Ovviamente, dopo un Else può esserci un altro if. In realtà, ciò accade molto spesso, perciò si trovano costrutti del tipo indicati in figura 3.

La "istruzione4" può essere considerata da eseguire per default, ovvero in tutte le condizioni eccettuate quelle definite dalle linee precedenti. Si noti che quando viene trovata un'espressione "vera", cioè diversa da zero, viene eseguita l'associata istruzione; in seguito l'esecuzione passa subito all'istruzione successiva all'intero gruppo (nel nostro caso specifico, a quella susseguente "istruzione4").

Un esempio banale potrebbe essere la funzione valuta() che restituisce zero se due numeri sono uguali, uno se il primo è maggiore del secondo, e -1 se è minore (figura 4).

In realtà, il linguaggio C consente anche un altro tipo di costrutto condizionale. Si basa sull'operatore ternario rappresentato dal punto di domanda (?) da utilizzare al posto di "if" in espressioni generali del tipo seguente:

(condizione) ? esprvera : esprfalsa

In questo caso, la condizione logica viene valutata e se risulta vera (diversa da zero), viene eseguita la singola istruzione "esprvera", altrimenti viene eseguita la singola espressione "esprfalsa".

Ad esempio, la funzione standard max(), che elabora il numero più grande tra i suoi due argomenti, viene implementata come una linea del tipo:

(a > b) ? a : b;

```
if (condizione) {
    blocco istruzioni 1
    }
else {
    blocco istruzioni 2
    }

(Figura 1)
```

```
if ( c == 'N' ) {
       printf("Fine");
       exit(0);
       }
else elabora();

(Figura 2)
```

```
if (espressione1)
    istruzione1
else if (espressione2)
    istruzione2
else if (espressione3)
    istruzione3
else
    istruzione4

(Figura 3)
```

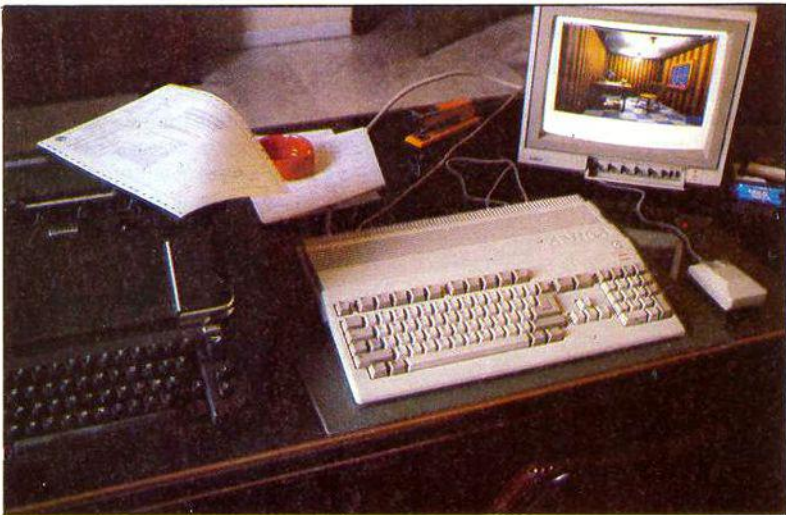
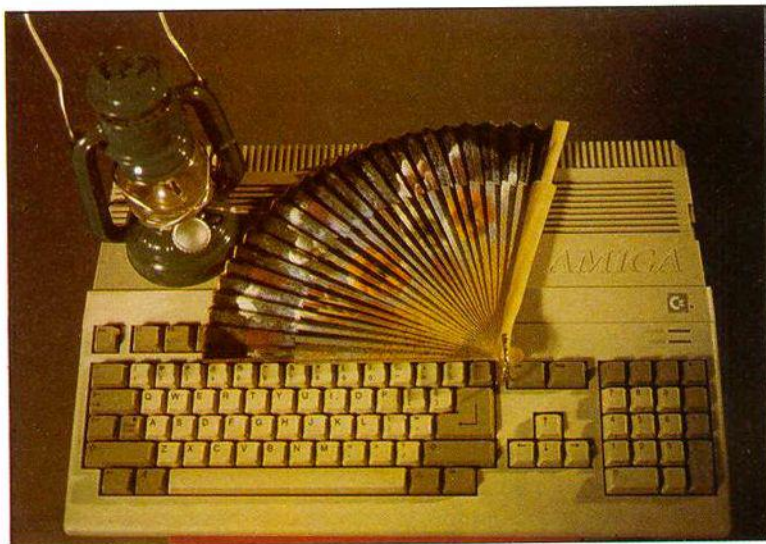
ISTRUZIONE SWITCH

Per chi conosce il Pascal, l'istruzione switch del C equivale a "case". Essa permette, cioè, di eseguire sequenze diverse di istruzioni a seconda del valore di una variabile di controllo.

La sintassi formale è indicata in figura 5.

Si noti che questa specifica, a differenza di IF-Else prevede già una serie di istruzioni come argomenti posti dopo case, perciò non servono le parentesi graffe. La sezione di default è opzionale.

L'istruzione switch valuta la variabili di controllo e la confronta con le costanti specificate dopo i case. Se si verifica una eguaglianza, viene eseguita l'appropriata sequenza di istruzioni sino ad un eventuale break, che passa l'esecuzione alla prima istruzione dopo la parentesi graffa di chiusura.

Se non viene trovata alcuna eguaglianza, viene eseguita la sequenza di istruzioni seguente default e, se

questa manca, il controllo passa alla prima istruzione dopo la graffa di chiusura senza che venga eseguito alcunchè.

L'applicazione tipica di switch è nella gestione delle opzioni, tipicamente degli input da tastiera. In figura 6 è riportata, ad esempio, una possibile sequenza di switch.

In questo caso il contenuto della variabile di carattere "ch" viene letto, e nel caso in cui contenga i, s, u vengono eseguite delle funzioni, altrimenti viene visualizzato il messaggio "Non valido".

CICLI DI ISTRUZIONI (LOOP)

La possibilità di ripetere cicli di istruzioni ad altissima velocità è una delle caratteristiche che sollevano il programmatore dal compito di eseguire calcoli ripetitivi o sequenze di operazioni sempre (quasi) uguali.

Il linguaggio C prevede alcune strutture di controllo di ciclo molto potenti.

Ad esempio, l'onnipresente ciclo for ha la sintassi formale indicata in figura 7.

Il primo parametro (inizial) specifica le condizioni di inizializzazione della/e variabile/i di ciclo. Il secondo parametro indica la condizione logica per cui il ciclo deve essere ripetuto. Il terzo parametro (incr) indica l'operazione di incremento da eseguire su di una variabile ad ogni esecuzione del ciclo (iterazione). Il blocco di istruzioni viene eseguito sinchè (cond) è vera, cioè diversa da zero.

Ad esempio, il seguente ciclo Basic...

FOR i=1 TO 100
*PRINT i, i*i*
NEXT i

...in linguaggio C potrebbe esser tradotto come in figura 8.

Si noti che tutti i parametri di for sono opzionali. Lo stesso programma avrebbe anche potuto scriversi, meno efficientemente ed elegantemente, come in figura 9.

Si noti che for permette di aggregare più valori di controllo, inizializzazione e incremento, come riportato in figura 10.

In questo caso, vengono inizializzate due variabili (i, j) rispettivamente a zero ed a sei (notare il separatore di virgola), poi il ciclo viene ripetuto sinchè i rimane inferiore a 20 ed anche (&&) j rimane maggiore di 5. Ad ogni chiusura di ciclo, viene eseguito un incremento di j di due).

Ultime due note: a differenza di molti interpreti Basic, il ciclo di istruzioni non viene eseguito nemmeno una volta se la condizione espressa in for non è vera sin dall'entrata. Inoltre, se non si vuole specificare uno dei parametri di for, bisogna comunque inserire il punto e virgola (devono essercene due, insomma, all'interno delle parentesi).

CICLO WHILE

Il ciclo while ha la sintassi formale indicata in figura 11.

Come sempre, se il "bloccoistruzioni" consiste di una sola istruzione,

```
int valuta(a,b)
{
if ( a == b ) return(0);
else if ( a > b ) return(1
else return (-1)
}
```

(figura 4)

```
switch (variab. controllo){
     case (costante1):
     sequenzaistruzioni1
     break;
     case (costante2):
     sequenzaistruzioni2
     break;
     .. .. ..
     default:
     sequenza default
     }
```

(Figura 5)

```
switch (ch) {
   case 'i':  istruzioni();
              break;
   case 's':  sort();
              break;
   case 'u':  exit(0);
              break;
   default:   printf("No");
   }
```

(Figura 6)

le graffe non sono necessarie. Il ciclo esegue un controllo del valore della condizione all'inizio, prima di eseguire le istruzioni; perciò se la condizione è falsa (pari a zero), non viene eseguito nulla ed il controllo passa alla prima istruzione seguente la graffa di chiusura (od il corpo della singola istruzione). La condizione può essere rappresentata da una espressione qualunque o da un aggregato di espressioni logiche collegate dagli opportuni operatori (vedere numero 51 di C.C.C.).

Notiamo subito che i due costrutti di figura 12 sono perfettamente equivalenti.

Ovviamente, il primo formato sarà più apprezzato dai programmatori FORTRAN e Basic ed il secondo dagli esperti di Pascal e Modula2, ma sono equivalenti in C.

LE SCELTE DI UN COMPUTER

Ciò che ad un computer conferisce la cosidetta intelligenza, se i filosofi consentono il pretenzioso termine, sono proprio le strutture di controllo del linguaggio di programmazione.

Infatti, tramite tali strutture, un elaboratore può affrontare un grande numero di situazioni diverse, modificando il proprio output e/o input in modo flessibile ed adeguato. Se non ci fossero parolette "condizionali", un programma potrebbe far fare ad un computer solo un ciclo infinito, e sempre uguale, di operazioni di I/O e difficilmente sarebbero affrontabili applicazioni di complessità più che elementare.

Il linguaggio C dispone di alcune strutture di controllo molto potenti, tipicamente pari o superiori a quelle di vari altri linguaggi evoluti. Inoltre, esse consentono di scrivere programmi ben strutturati e senza fare uso dei tanto odiati "salti" (GoTo) che viziano i programmi Basic e Fortran, diminuendo l'efficienza dell'elaboratore e rendendo poco efficienti i programmi stessi.

CICLO DO-WHILE

Esiste un tipo di ciclo apparentemente gemello di WHILE, ma assai diverso.

La sintassi formale è:

do istruzione
while (espressione)

In questo caso l'istruzione (od il blocco racchiuso tra graffe), viene eseguita sempre almeno una volta prima di verificare la condizione. Se questa è vera, allora l'istruzione viene ripetuta, altrimenti l'esecuzione passa alla linea successiva a while.

Questo tipo di ciclo è molto utile per i viziati dal Fortran e Basic, in quanto il classico schema di programmazione (e mentale) tipo...

100 istruzione
200 If (cond) THEN GoTo 100

...tanto usato con questi linguaggi, può essere direttamente convertito.

Si noti che la istruzione che segue "do" viene sempre eseguita almeno una volta, prima di eseguire il controllo della condizione.

Ad esempio, vediamo un ciclo che legge un carattere (funzione getchar()) sinchè non è uguale a "s" oppure "n" (figura 13).

BREAK e CONTINUE

In C esistono due istruzioni che permettono di "uscire" da un ciclo senza ricorrere al famigerato GoTo. Sia Break che Continue non hanno parametri, ma agiscono in modo piuttosto dissimile.

Indicando break in un ciclo (tra le graffe dopo do, while, for), l'esecuzione passa alla prima specifica successiva all'ultima istruzione del ciclo attuale.

Supponendo di avere una funzione chiamata tasto() che restituisce un valore vero (diverso da zero) quando si pigia un tasto, una funzione che permette di uscire da un ciclo di cento iterazioni potrebbe essere quella indicata in figura 14.

In questo caso, se l'utente pigia un tasto, il ciclo non viene eseguito cento volte, ma subito si passa all'esecuzione della prima istruzione dopo le graffe di chiusura.

```
or ( inizial; cond; incr ){
  blocco di istruzioni
 }
Figura 7)
```

```
nt i;
or ( i=1; i<=100; i++ )
   printf("n\%d %d",i,i*i);
Figura 8)
```

```
nt i=1;
or ( ; i<=100 ; ) {
 printf("n\%d %d", i, i*i);
 i++;
 {
Figura 9)
```

```
for ( i=0,j=6; i<20 && j>5 ; j+=2){
.. istr .. ..
}
(Figura 10)
```

```
while (condizione) {
        bloccoistruzioni
}
```

(Figura 11)

```
for ( e1; e2; e3)
     istruzione

e1;
while ( e2 ){
     istruzione;
     e3;
}
```

(Figura 12)

```
char c;
do
   getchar(c);
while ( c!= 'n' && c!= 's')
```

(Figura 13)

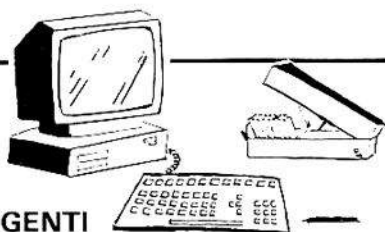
```
while ( x < 101 ){
    .. istruzioni..
    if (tasto()) break;
    .. istruzioni..
}
```

(Figura 14)

```
for (x=1;x<=100;x++ ){
   .. istruzioni1 ..
   if (tasto()) continue;
   .. istruzioni2 ..
}
```

(Figura 15)

```
while ( x<=100){
 ..istruzioni..
 if (tasto()) goto etic;
 .. istruzioni..
}
etic:
..istruzioni..
```

(Figura 16)

EDITOR E TESTI SORGENTI

Gli interpreti Basic in circolazione, compresi quelli di C/64 e Amiga, solitamente riconoscono la fine di una linea di programma tramite un codice di Newline oppure di Carriage-Return (rispettivamente, 10 e 13 in ASCII).

Poichè tali caratteri di controllo non sono stampabili, spesso ci si ritrova con programmi che funzionano in modo inspiegabile, pur essendo il listato perfetto, in quanto due o più linee risultano "fuse" tra loro perchè non si è pigiato il tasto Return tra l'una e l'altra. Al momento dell'interpretazione il computer fa confusione (trova un numero di linea dopo un argomento di PRINT eccetera) ed emette, molto spesso, un Syntax error.

Il linguaggio C, invece, assume come terminatore di istruzione il carattere di punto e virgola (;). Ciò significa che, quando si scrive una linea molto lunga od articolata, si può tranquillamente premere il Return per andare a capo e proseguire. Solitamente, la lunghezza massima di una singola linea di istruzioni, anche se effettivamente giace su più linee del testo sorgente, è complessivamente di 255 caratteri sui piccoli personal.

Continue, invece, provoca all'interno di un ciclo la valutazione immediata della successiva iterazione. Ovvero, è come se ci fosse un salto alle parentesi graffe di chiusura (non dopo). Nel caso di figura 15, quando l'utente pigia un tasto, il gruppo di istruzioni 2 non verrà eseguito, ma verrà subito ripetuto il gruppo di istruzioni 1. In pratica, il primo gruppo di istruzioni verrà eseguito sempre 100 volte, il secondo 100 volte meno il numero di volte che l'utente pigia un tasto durante l'esecuzione del ciclo.

GOTO e LABEL

L'istruzione di salto è odiata da tutti i cultori della programmazione strutturata, ma è fornita, fosse solo per dare sicurezza ai programmatori Fortran e Basic che invece l'amano al punto da non riuscire a vivere senza (ma devono imparare a tradirla in C!).

Sebbene vi siano programmatori C che in vent'anni di carriera non hanno mai usato un GoTo, possono verificarsi particolari condizioni in cui esso è quasi indispensabile, pena dure riscritture di molte linee del sorgente. La sintassi del salto è simile all'AmigaBasic. Riproponendo l'esempio visto per break, un suo uso può esser rappresentato dall'esempio di figura 16.

Si noti il carattere di doppio punto (:) dopo il nome dell'etichetta che identifica la sezione del programma a cui si vuole saltare. Più avanti vedremo le limitazioni di tale istruzione in ambienti multiprogrammabili come Amiga.

A SCUOLA CON UN COMPUTER

L'esperienza di un maestro elementare alle prese con un allievo elettronico piuttosto particolare...

di **Tiziano Trivella**

Faccio il maestro elementare in una piccola scuola di collina organizzata a pluriclassi. Da tre anni ho introdotto il computer in aula e, soprattutto, lo faccio usare ai miei alunni a scopo educativo e didattico.

Specifico subito che non sono uno specialista di informatica, come, del resto, non sono un musicista anche se insegno musica, nè un meteorologo anche se analizzo, con i bambini, i fenomeni atmosferici. Questo lo dico perchè spesso gli educatori (genitori ed insegnanti) credono che per insegnare occorra necessariamente essere degli "specialisti". Per quanto mi riguarda ho constatato in varie occasioni che, di norma, gli specialisti sono troppo esperti nella loro materia per poterla insegnare a bambini di 8 o 9 anni, cioè troppo sicuri e troppo poco dubbiosi per poter capire i dubbi e le incertezze di uno scolaro in fase di apprendimento.

Quindi ritengo che un insegnante (e particolarmente un maestro elementare) debba, sì, conoscere la "materia" che si appresta ad insegnare, ma soprattutto saperne ricavare le valenze educative in rapporto ai fini che si prefigge e, sempre, in riferimento alle persone (gli alunni)

che ha davanti, se veramente vuole contribuire a sviluppare le loro potenzialità ed aiutarli a crescere culturalmente.

Certo, il difficile è proprio questo: ricavare dalla realtà le valenze educative, cioè leggere ed interpretare i fatti, i fenomeni e le cose per costruire in seguito griglie di comprensione da proporre ai bambini ed ai giovani, per costruire la cosiddetta "cultura".

Sicuramente è più facile la politica dello struzzo: negare i fenomeni reali e far finta di niente. Appunto come chi afferma che il computer rovina i giovani e, in base a questa affermazione gratuita, ne sconsiglia o ne proibisce l'uso domestico o scolastico. Stiamo organizzando un pianeta telematico e... vogliamo impedire ai nostri figli di conoscere e usare un computer?

Comunque sappiamo tutti che, storicamente, le battaglie contro i progressi tecnologici sono già perse in partenza: i cavalli, le candele, la carta e le matite esistono ancora, ma noi usiamo le automobili, le lampadine ed i computer.

Di conseguenza è inutile polemizzare sull'argomento; è più utile e necessario, al contrario, comprendere e valorizzare le potenzialità delle tecnologie avanzate, iniziando proprio dalla scuola. Con semplicità, ma con chiarezza.

Fino a tre anni fa ero completamente digiuno di informatica e di computer (non ne avevo mai toccato uno!): ma ero molto ben disposto a cercare di capirci qualcosa. Un bel giorno, in accordo con i genitori che gestivano una cassa scolastica volontaria, decidemmo di rivendere un proiettore superotto acquistato l'anno precedente per comperare un computer. L'occasione mise sulla nostra strada un Commodore Plus/4 nuovo (ma già vituperato dai più e, forse, dalla Commodore stessa). Comunque per noi si dimostrò un buon affare didattico.

Per un giorno lo tenni a casa mia per imparare almeno ad effettuare i necessari collegamenti tra fili e prese e per provare a scrivere Print "Salve ragazzi!", altrimenti... che figura avrei fatto con i miei stessi alunni?

Quel giorno in classe l'emozione fu grande: nessun altro sussidio didattico aveva mai provocato così viva curiosità ed interesse tra i bambini, nessuno escluso. E l'interesse, si sa, è sempre un buon punto di partenza.

Col passare del tempo i miei alunni ed io imparammo ad usare il computer, comando dopo comando e istruzione dopo istruzione, seguendo il manuale ed alcune riviste del settore (tra cui C.C.C.). In seguito predisposi una programmazione didattica nella quale fissai gli obiettivi da raggiungere. Ritengo necessario riportare la premessa alla programmazione didattica impostata:

"Nella coscienza che l'informatica, al di là degli aspetti consumistici e di moda, è una realtà ormai presente in modo definitivo in molti aspetti della nostra vita, si ritiene opportuno e possibile sperimentare, anche nella scuola dell'obbligo, un approccio a livello elementare a tale scienza, vista la valenza formativa di cui essa è portatrice in ordine a:

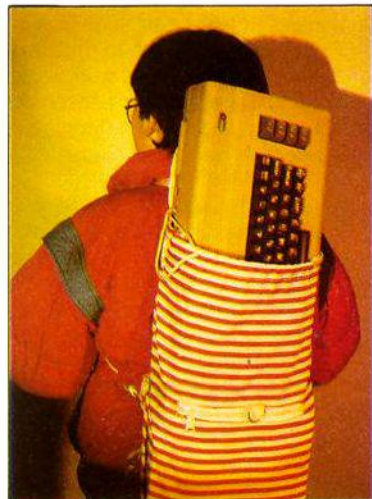
- *la struttura estremamente logica del "linguaggio".*
- *le sue possibilità applicative interdisciplinari.*

Per quanto riguarda il primo punto, si sottolinea che i linguaggi utilizzati in informatica costringono l'alunno a chiarire e definire i passi logici attraverso i quali si esplica un processo di ragionamento e a tradurne le procedure in istruzioni per l'elaboratore.

In riferimento al punto successivo, invece, si evidenzia che l'informatica, per la sua struttura concettuale, rientra direttamente, senza forzature, nelle aree logico-matematica e scientifica. Inoltre il computer può essere un valido sussidio didattico anche per tutte le altre materie tradizionali del curriculum scolastico come, ad esempio: archivio dati per geografia, analisi linguistica, geomtria, storia; per elaborazioni grafiche, per musica, archivio biblioteca, prove di verifica strutturate, memorizzazione dati per la valutazione degli alunni e così via."

Riuscimmo, con il passar del tempo, a raggiungere gli obiettivi specifici prefissati, tramite un processo continuo di lavoro che permise (a me ed ai bambini) di evidenziare che:

- il computer è solo una scatola vuota, in assenza dell'operatore che lo sappia guidare e utilizzare.
- il computer è un sussidio didattico di tipo "attivo", nel senso che l'alunno ne è l'operatore e interagisce con esso: senza operatore il computer rimane come un soprammobile inutile.
- il computer costringe l'alunno a ricercare gli errori e a rivedere le proprie procedure di ragionamento (ma, nello stesso tempo, non rimprovera nè punisce...)
- le straordinarie capacità della macchina non servono a nulla, senza le capacità della persona che la usa.
- il computer è un mezzo di lavoro, uno strumento e, magari, anche un gioco, non un oggetto magico nè un mito.

Le soddisfazioni più belle sono venute quando i bambini sono stati in grado di produrre brevi e semplici

programmi, totalmente creati da loro, in base alle conoscenze acquisite: allora si sono veramente sentiti padroni della macchina.

Un alunno ha scritto: "Il computer mi ha aiutato ad essere più intelligente e preciso"; ed un'alunna: "Il computer mi è stato utile per sviluppare la fantasia inventando programmi".

Direi che queste motivate affermazioni possono, da sole, dare chiare indicazioni sull'uso educativo del computer.

Prima di concludere vorrei sottolineare (visto che questa è una rivista per utenti Commodore) che anche il semplice Plus/4 si è rivelato molto utile ad un'esperienza scolastica, come lo può forse essere il C/16, per la facilità di gestione della grafica e del sonoro, per non parlare del w/p incorporato da noi utilizzato spesso da quando abbiamo anche la stampante.

Se tra i molti lettori di questa rivista vi sono altri insegnanti interessati ad uno scambio di opinioni e di esperienze, il discorso si potrebbe allargare e farsi anche più specifico: ad esempio con la presentazione di singole unità didattiche e di programmi prodotti dagli insegnanti e dagli alunni.

L'importante è fare in modo che il computer non diventi un soprammobile in più nelle aule scolastiche o, peggio, che ci si senta intimoriti e "bloccati" solo perchè inesperti: insegnanti ed alunni lo devono toccare, provare ed usare in prima persona. Solo così riusciremo a smitizzare questo prodotto di alta tecnologia, rapido nei calcoli, preciso nelle elaborazioni e potente nella memoria, ma debole al punto che se stacchi la presa di corrente non serve più a niente e che il cervello e la mano di un bambino lo possono comandare con facilità.

Tiziano Trivella
Scuola elementare
"Roncallo Gaggio"
24030 Pontida (Bg)

Un laboratorio di logica

L'intervento del nostro lettore dimostra, ancora una volta, che un calcolatore è lo strumento ideale per far capire ad un giovane allievo la "logica", in senso lato.

Come giustamente viene sottolineato, "un computer non rimprovera e non punisce" ma stimola certamente, in modo costruttivo, a darsi da fare per ricercare gli errori commessi. Quanti studenti, ancor oggi, preferiscono copiare da un compagno il compito assegnato per casa e che non hanno avuto voglia di svolgere!

E' molto più difficile, invece, che uno studente rinunci a cercare, da solo, la causa della segnalazione di errore, e ve ne sono tante, soprattutto all'inizio!

Il computer rappresenta, insomma, una sfida vera e propria, magari a livello inconscio, tra uomo e macchina; illuminante, infatti, la domanda che gli studenti pongono, in caso di errore, dopo le prime applicazioni ("Perchè il computer sbaglia il risultato?") e dopo qualche tempo ("Dove ho commesso l'errore segnalato?").

Ciò dimostra, quanto meno, un nuovo atteggiamento di una specie di umiltà, "pefino" di fronte ad una macchina che, pur se "stupida", sembra in grado di accorgersi dei nostri errori e di segnalarli, pur se con la dovuta discrezione.

Il rigore della logica, insomma, prende il sopravvento dopo le prime esperienze; gli atteggiamenti istintivi lasciano il posto a quelli più riflessivi, logici, appunto.

Ciò che dispiace, invece, è la condizione di una gran parte delle scuole italiane, in cui, per andare avanti, si è costretti a vendere uno strumento didattico per procurarsene ùn altro.

Ci domandiamo, allora, che cosa accadrebbe se gli insegnanti di buona volontà, come Trivella, potessero mettere, a disposizione della didattica, non gli spiccioli faticosamente raccolti da genitori e professori (per far fronte alle note, gravi carenze "ufficiali"), ma quei milioni che, invece, prendono sempre più spesso la strada di misteriose valigette con destinazioni tutt'altro che didattiche...

A.d.S

ORIGINALE O COMPATIBILE, QUESTO E' IL DILEMMA

I drive 1541 compatibili stanno conquistando l'Italia; ma il compatibile conviene oppure no?

di **Fabio Sorgato**

Ormai il mercato dei computer professionali è invaso da compatibili di tutte le marche, colori e prezzi; classico è l'esempio dei computer che girano con il sistema operativo Ms-Dos, i cosiddetti IBM compatibili.

La forte richiesta di elaboratori a basso costo, grazie anche alla ricca biblioteca di software disponibile, ha spinto una folto gruppo di piccole aziende del sol levante (soprattutto Taiwan) a gettarsi a capofitto nella produzione di cloni iper-economici, creando una concorrenza spietata e un giro d'affari di miliardi di dollari.

Tempo fa qualcuno di questi orientali ha capito che, tra gli utenti della fascia degli home computer, c'era un esercito di studenti, professionisti e casalinghe che, possessori di Commodore 64, chiedevano disperatamente la tanta sospirata unità a disco ad un prezzo stracciato: sono nati così (più o meno) i drive compatibili 1541.

SCHEDA TECNICA

Software applicativo per test di velocità su drive

Hardware richiesto: C/64, C/16, Plus/4, C/128, Vic-20; non adattabile ad Amiga

I programmi pubblicati sono specifici per disk drive e sono consigliati anche a coloro che già posseggono un drive

Anche i programmi pubblicati in queste pagine sono contenuti nel disco "Directory" di questo mese.

Anche noi ci siamo fatti prendere dalla curiosità per questi gioielli dell'informatica; daremo il nostro piccolo contributo, in questo articolo, per aiutare chi si appresta ad acquistare il suo primo (o secondo) drive.

Stiamo parlando dei due drive RF-501 e RF-531 prodotti in Taiwan da un'azienda che fabbrica anche drive IBM compatibili, Amiga compatibili e perfino un drive compatibile 1571 (l'esperienza a loro non manca!); la scelta è caduta su questi drive per due motivi principali: sono prodotti facilmente reperibili ed affidabili.

RF 501 & RF 531

Quali sono le differenze tra i due drive? La risposta è semplice: nessuna differenza di rilievo, ovvero la sola ed unica diversità risiede nell'involucro esterno che, nel caso del 501, è in materiale plastico, mentre nel 531 è in metallo; inoltre quest'ultimo vanta anche dimensioni leggermente più ridotte grazie alla maggiore dissipazione di calore di cui il metallo è capace (ingombri: 27x15x4.5 per il 531, 27x17x7 per il 501 e 37.4x20x9.7 per il 1541 della Commodore).

Il prezzo di mercato dei due prodotti è sostanzialmente identico; eventuali differenze di costo vengono assorbite dai rivenditori. Non ci sono comunque motivi per cui si debba preferire un 531 a un 501: scegliete tranquillamente quello che vi piace di più.

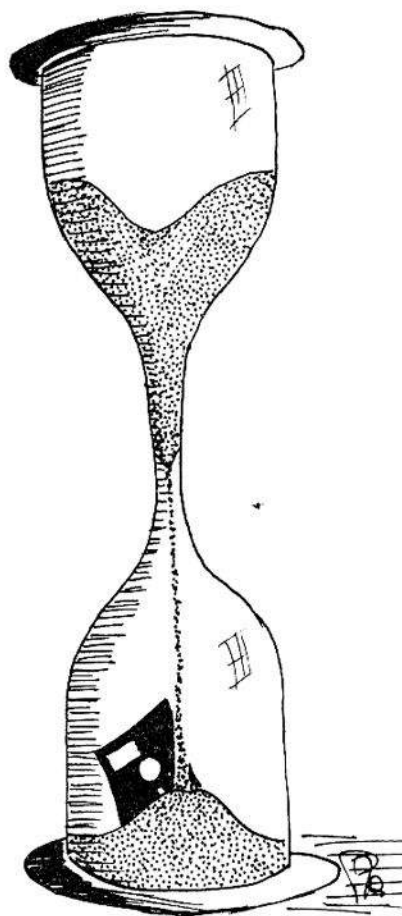
Nella confezione è presente il drive, in cui è opportunamente inserito il classico protection-sheet, l'alimentatore ed il cavo di connessione per la porta seriale.

I drive, come le altre apparecchiature elettroniche, godono di un periodo di garanzia, che varia a seconda dell'importatore e del distributore; scaduto tale periodo si pone il problema di eventuali riparazioni, ma abbiamo appreso con soddisfazione che molti centri di assistenza Commodore (e non) sono in grado di riparare tali periferiche dal momento che sono copie (quasi) gemelle dei noti 1541.

Compatibilità

La nostra prima curiosità è stata quella di verificare se realmente la velocità dei drive fosse superiore a quella del 1541 originale.

Con tre semplici test abbiamo verificato la velocità di spostamento della testina, di accesso alla traccia, la velocità di scrittura e di lettura di dati sequenziali: non c'è nessuna differenza apprezzabile nei tempi rilevati, ciò a confermare un funzionamento del tutto simile al 1541 e quindi una elevata compatibilità.

Anche nelle prove con i copiatori gli esemplari provati se la sono cavata egregiamente; la prima prova è stata eseguita con FCOPY 1.5, un copiatore, per chi non lo conoscesse, in grado di eseguire il backup di un dischetto in meno di due minuti. Le copie sono riuscite perfettamente e gli stessi risultati li abbiamo avuti con il DOUBLE-IMAGE, un nibble-copy in grado di copiare dischetti

```
100 REM ** BENCHMARK PER DRIVE 1541/1571
110 REM ** ORIGINALI E COMPATIBILI
120 :
130 REM ** BY FABIO SORGATO
140 :
150 REM ** TEST DI MOVIMENTO TESTINA
160 REM ** E ACCESSO ALLA TRACCIA
170 :
180 OPEN 5,8,5,"#":REM CANALE-BUFFER
190 OPEN 15,8,15:REM CANALE COMANDI
200 FOR K=1 TO 35
210 REM SPOSTA TESTINA SULLA TRACCIA K
220 PRINT# 15,"B-R:";5;0;K;0
230 REM SPOSTA TESTINA SU TRACCIA 36-K
240 PRINT# 15,"B-R:";5;0;36-K;0
250 NEXT
260 CLOSE 5:CLOSE 15:REM CHIUDE CANALI
270 :
280 REM TEMPO MEDIO 1541:  39 SECONDI
290 REM TEMPO MEDIO 1571:  35 SECONDI
```

protetti; nessun problema è stato riscontrato con altri copiatori file-file.

Il drive è stato anche in grado di caricare programmi "difficili" come il Geos, Easy Script (versione originale), Koalapainter e vari giochi protetti che si trovano comunemente in commercio.

Il funzionamento è silenzioso e la macchina non riscalda neppure dopo moltissime ore (un giorno, per errore, abbiamo dimenticato di spegnerlo...)

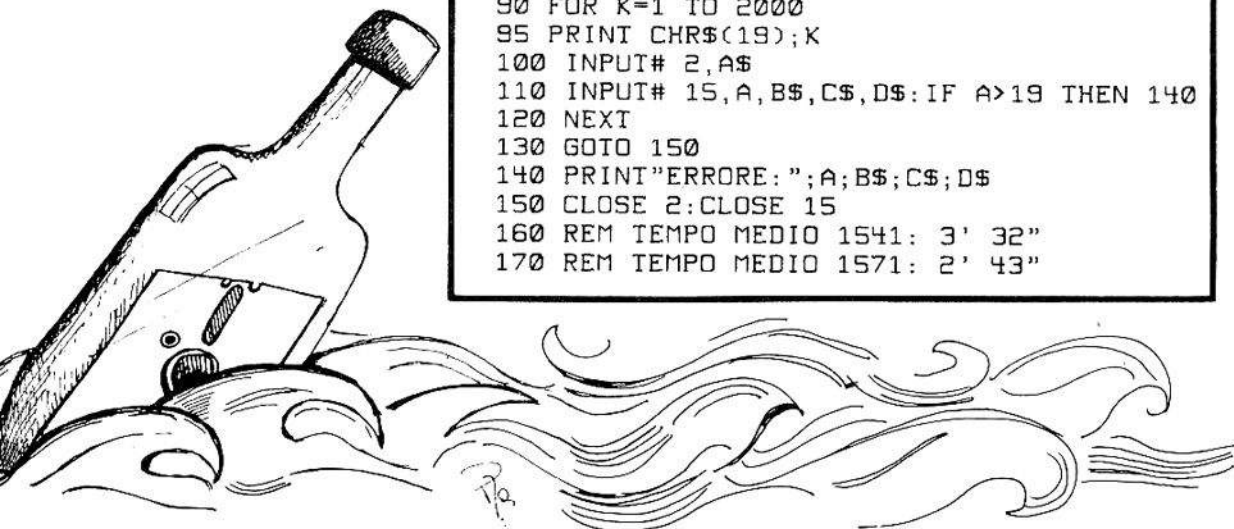
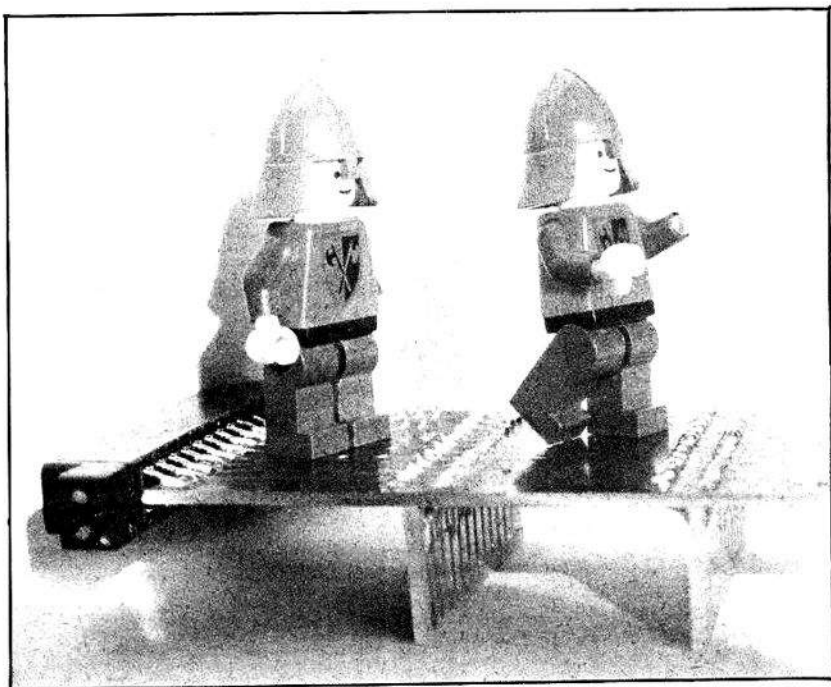
I Benchmark

Sono qui pubblicati i tre minilistati sviluppati per effettuare un confronto tra l'originale e il compatibile; le prove, come anticipato, sono di accesso alla traccia e spostamento testina, scrittura e lettura: i risultati sono rigorosamente identici per i tre modelli (1541, RF501, RF 531).

Anche la velocità di lavoro con i file relativi risulta invariata sulle due apparecchiature.

Non abbiamo riscontrato nessuna incompatibilità durante le nostre prove e riteniamo che non ve ne siano; è comunque possibile che con l'uso di copiatori particolarmente sofisticati e veloci sorgano difficoltà dovute alla seppur minima differenza di "costituzione"; caso mai sorgessero problemi il rimedio è semplicissimo: cambiate copiatore!

```
100 REM ** BENCHMARK N.2
110 REM ** TEST 1541 E COMPATIBILI
120 REM ** BY FABIO SORGATO
130 :
140 REM ** SCRITTURA DI 2000 DATI
150 REM ** SEQUENZIALI
160 :
170 PRINT CHR$(147) TAB(10)"-ESIMO DATO"
180 OPEN 15,8,15
190 PRINT# 15,"S0:PROVA,S"
200 OPEN 2,8,2,"PROVA,S,W"
210 FOR K=1 TO 2000
220 PRINT CHR$(19);K
230 PRINT# 2,"C.C.C."+STR$(K)
240 INPUT# 15,A,B$,C$,D$
250 IF A>19 THEN 280
260 NEXT
270 GOTO 290
280 PRINT "ERRORE:";A;B$;C$;D$
290 CLOSE 2:CLOSE 15
300 REM TEMPO MEDIO 1541: 3' 58"
310 REM TEMPO MEDIO 1571: 3' 33"
```

```
10 REM ** BENCHMARK N.3
20 REM ** TEST 1541/1571 E COMPATIBILI
30 REM ** BY FABIO SORGATO
35 :
36 REM ** PROVA DI LETTURA
37 REM ** DI 2000 DATI SEQUENZIALI
40 :
45 PRINT CHR$(147)TAB(10)"-IESIMO DATO"
50 OPEN 15,8,15
70 OPEN 2,8,2,"PROVA,S,R"
90 FOR K=1 TO 2000
95 PRINT CHR$(19);K
100 INPUT# 2,A$
110 INPUT# 15,A,B$,C$,D$:IF A>19 THEN 140
120 NEXT
130 GOTO 150
140 PRINT"ERRORE:";A;B$;C$;D$
150 CLOSE 2:CLOSE 15
160 REM TEMPO MEDIO 1541: 3' 32"
170 REM TEMPO MEDIO 1571: 2' 43"
```

UNO SPRITE INDELEBILE

A volte può essere utile riportare su carta gli sprite realizzati con tanta fatica

di **Giancarlo Mariani**

Dall'uscita del Commodore 64, fino ad ora, sono stati sviluppati moltissimi programmi che consentono di sfruttare al meglio le possibilità di questo computer, in special modo per quanto riguarda suono e grafica (e quindi sprite), ma non sono state studiate routine che consentano di stampare su carta questi ultimi.

La routine di cui parliamo è naturalmente quella proposta in queste pagine: permette la stampa di uno sprite posto a partire da una qualsiasi locazione di memoria. E' stata progettata per le stampanti del tipo 801/803 e compatibili varie.

La spiegazione completa del suo tunzionamento non è delle più semplici; vi suggeriamo, quindi, di esaminare con attenzione anche le varie tabelle ed programmi pubblicati.

Sprites (27707 / 27982)

Per chi non lo sapesse ancora(!), ricordiamo che uno sprite è formato da una matrice di 24 punti (o pixel) orizzontali x 21 verticali e può essere rappresentato mediante una tabella. Dato che ogni byte è formato

da 8 bit, avremo, su ogni riga dello sprite, 3 byte di memoria, e, dato che le righe sono 21, risulta che per memorizzare uno sprite sono necessari 21x3=63 byte. La memoria del C/64, che è di 64k, ne può contenere, quindi, 1024.

La stampante, purtroppo, ha 7 aghi disposti, per di più, in verticale: bisognerà, pertanto, leggere i dati di uno sprite in modo insolito prima di mandarli in stampa.

In pratica, bisogna considerare dapprima i primi 7 punti verticali nella prima colonna della tabella ed inviarli alla stampante; quindi passare ai successivi 7 punti, poi alla nuova fila, e così via fino ai 24-esimi.

Dopo aver fatto questo bisogna inviare un Carriage Return [Chr$(13)] e leggere i 7 punti verticali dal n.8 al n.14 più a sinistra, per poi continuare, in orizzontale, fino ai 24-esimi. Stessa cosa si fa per gli ultimi 7 punti verticali (dal n.15 al n.21).

Immaginate, insomma, che un byte sia composto da 7 bit verticali, e che il primo byte da inviare alla stampante sia quello posto più in alto a sinistra della tabella, mentre l'ultimo è quello più in basso a destra. Il senso di scorrimento è, ovviamente, da sinistra a destra e dall'alto al basso.

Questa spiegazione, tradotta in programma, è molto meno semplice di quello che potrebbe sembrare: ve ne accorgerete, infatti, esaminando il disassemblato della routine; chi non riuscisse a comprenderne il funzionamento può esaminare il listato Basic allegato che è lo stesso programma, "tradotto" però in Basic.

Le varie operazioni tipo AND, elevamenti a potenza o cose del genere, presenti in entrambi i listati, servono per verificare se il punto controllato è acceso o spento e per formare i byte da inviare alla stampante.

La sintassi della routine è la seguente:

Sysxxxx, Spr [,bank [,device]]

in cui XXXX è l'indirizzo di partenza della routine (qualsiasi, dato che la routine è rilocabile).

SPR è il numero dello sprite da stampare, compreso tra 0 e 255.

BANK è il banco di memoria tra 0 e 3. Questo parametro è opzionale e se viene omesso verrà preso di default BANK=0 (locazioni tra 0 e 16383).

DEVICE è il numero della periferica alla quale andranno inviati i dati. Il parametro è opzionale, e se viene omesso verrà preso, di default, DEVICE=4.

La sintassi usata nei casi più semplici sarà:

Sys xxxx, Spr

Per trovare la locazione di partenza in cui è memorizzato lo sprite da stampare, deve essere usata la seguente formula:

```
Come e' Fatto uno sprite


     12345678    12345678    12345678
     ----------------------------------
  1  ........ +  ........ +  ........
  2  ........ +  ........ +  ........
  3  ........ +  ........ +  ........
  4  ........ +  ........ +  ........
  5  ........ +  ........ +  ........
  6  ........ +  ........ +  ........
  7  ........ +  ........ +  ........
------------+-----------+-------------
  8  ........ +  ........ +  ........
  9  ........ +  ........ +  ........
 10  ........ +  ........ +  ........
 11  ........ +  ........ +  ........
 12  ........ +  ........ +  ........
 13  ........ +  ........ +  ........
 14  ........ +  ........ +  ........
------------+-----------+-------------
 15  ........ +  ........ +  ........
 16  ........ +  ........ +  ........
 17  ........ +  ........ +  ........
 18  ........ +  ........ +  ........
 19  ........ +  ........ +  ........
 20  ........ +  ........ +  ........
 21  ........ +  ........ +  ........
     ----------------------------------
```

Loc=16384 * Bank + 64 * Spr

In pratica, con SPR si riesce a spaziare in 16k di memoria, a gruppi di 64 byte alla volta; fate quindi attenzione a posizionare gli sprite a partire da locazioni divisibili esattamente per 64, altrimenti lo sprite non verrà stampato correttamente.

BANK definisce quale banco di memoria da 16k usare. Le zone di "esplorazione" saranno le seguenti:

BANK=0: tra 0 e 16383
BANK=1: tra 16384 e 32767
BANK=2: tra 32768 e 49151
BANK=3: tra 49152 e 65535

Per fare un esempio, supponendo di assegnare...
SYS xxxx,20,1

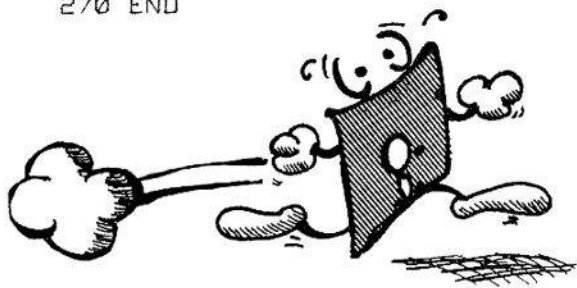
...la locazione di partenza dello sprite sarà:

20*64+1*16384=17664

Prestate molta attenzione al fatto che la routine l.m. utilizza le locazioni da $F7 a $FF, e quindi cercate di non allocare sprite nelle suddette celle di memoria, altrimenti vi lascio immaginare cosa succede.

La versione Basic, che è esattamente la traduzione del programma l.m., è sconsigliabile da usare, per la sua estrema lentezza, ma può aiutare a capire, insieme al disassemblato commentato, il funzionamento della routine l.m.

Ricordo che non possono essere stampati gli sprite definiti in multicolor.

```
  2 REM  ** STAMPA SPRITE
  3 REM  ** VERSIONE BASIC
  4 REM  ** BY MARIANI G.
  5 :
 10 PRINT CHR$(147)
 20 DEV=4:BANK=0:REM  ** DEFAUL
    TS
 30 INPUT "NUMERO SPRITE ";NS
 35 IF  NS<0  OR  NS>255  THEN
     30
 40 INPUT "BANCO (CR=0) ";BANK
 50 IF  BANK<0  OR  BANK>3  THE
    N  40
 60 INPUT "PERIFERICA (CR=4) ";
    DEV
 70 :
100 P=(NS*64)+(16384*BANK):REM
     ** INDIR. INIZIALE ($FE/$F
    F)
105 :
110 OPEN  1,DEV:PRINT#1,CHR$(8)
120 FOR  K=0  TO  2:REM  ** $FA
130 FOR  L=0  TO  2:REM  ** $FB
140 FOR  B=7  TO  0  STEP  -1:R
    EM  ** $FC
150 V=128:REM  ** $02
160 FOR  Q=0  TO  18  STEP  3:R
    EM  ** $FD
170 X=P+(Q+L+K*21):REM  ** IND.
     CARICAMENTO
180 G=-((PEEK(X) AND (2↑B))=(2↑
    B))
190 V=V+G*2↑(Q/3):REM  ** CARAT
    TERE DA STAMPARE
200 NEXT Q
210 PRINT#1,CHR$(V);:REM  ** ST
    AMPA
220 NEXT B
230 NEXT L
240 PRINT#1:REM  ** LINEA VUOTA
250 NEXT K
260 PRINT#1,CHR$(15):CLOSE 1
270 END
```

Per chi inizia

Ricordiamo, ai nuovi lettori che ci leggono per la prima volta, che la nostra rivista propone su ogni numero un gruppo di routine in Linguaggio Macchina, per il Commodore 64, interamente rilocabili. Ciò significa che l'utente potrà realizzare una vera e propria enciclopedia in L.M. personalizzata in base alle proprie esigenze.

E' ovvio che le routine proposte, per funzionare adeguatamente, soggiacciono ad alcuni limiti, ben illustrati nei primi numeri di "Nuovo Sistema".

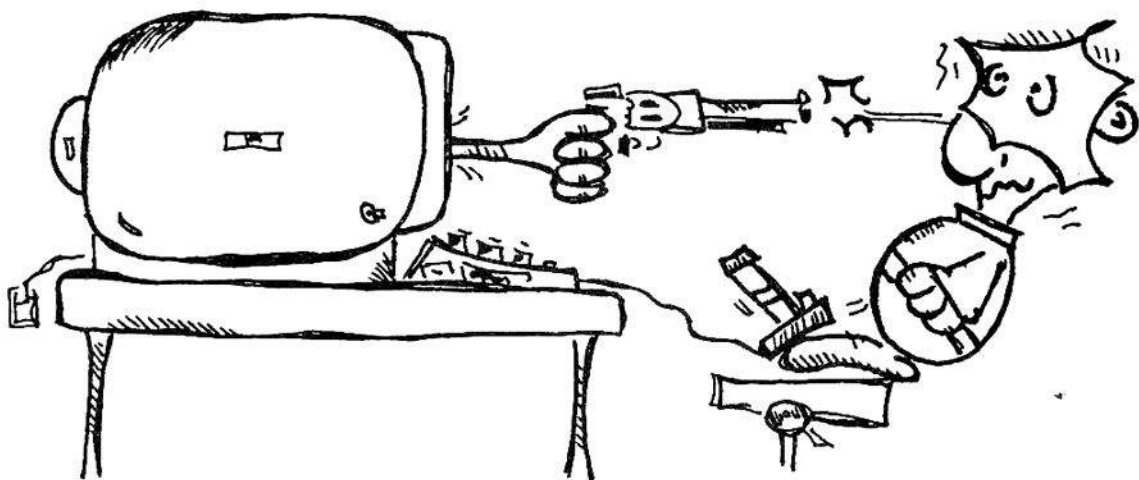
In questa sede ci limitiamo a ricordare che, per evitare malfunzionamenti, è necessario:

- Fissare il Top di memoria Ram a 20000.
- Caricare, servendosi del programma "Caricatore" a suo tempo pubblicato (o analoghi), le routine Basic contenenti le istruzioni Data.
- Indicare la prima locazione di memoria in cui si desidera allocare la routine stessa.
- Attivare la routine secondo i suggerimenti indicati nei Demo a corredo.

```
1000 PRINT CHR$(147);"STAMPA SPR
     ITE":PRINT
1010 PRINT"USO: SYS XXXX,NS [,BA
     NK [,DEVICE ]]":PRINT
1020 PRINT"XXXX=INDIRIZZO DI PAR
     TENZA QUALSIASI":PRINT
1030 PRINT"NS=NUMERO SPRITE (0..
     255)":PRINT
1040 PRINT"BANK=BANCO (0..3). OP
     ZIONALE: DEFAULT=0":PRINT
1050 PRINT"DEVICE=PERIFERICA. OP
     ZIONALE: DEFAULT=4"
1060 RETURN
1070 :
1100 DATA  169,000,133,247,169,0
     04,133,248,032,253,174,032
1110 DATA  158,183,169,000,133,2
     50,134,249,160,006,006,249
1120 DATA  038,250,136,208,249,1
     69,044,160,000,209,122,208
1130 DATA  031,032,253,174,032,1
     58,183,224,004,144,003,076
1140 DATA  072,178,134,247,169,0
     44,160,000,209,122,208,008
1150 DATA  032,253,174,032,158,1
     83,134,248,165,249,133,254
```

```
1160 DATA  165,247,010,010,010,0
     10,010,010,216,024,101,250
1170 DATA  133,255,169,187,166,2
     48,160,000,032,186,255,169
1180 DATA  000,032,189,255,032,1
     92,255,162,187,032,201,255
1190 DATA  169,008,032,210,255,1
     69,000,133,250,169,000,133
1200 DATA  251,169,007,133,252,1
     69,128,133,002,169,000,133
1210 DATA  253,165,250,024,162,0
     20,101,250,202,208,251,101
1220 DATA  251,101,253,166,255,0
     24,101,254,133,248,144,001
1230 DATA  232,134,249,160,000,1
     69,001,133,247,166,252,240
1240 DATA  005,006,247,202,208,2
     51,177,248,037,247,197,247
1250 DATA  208,032,162,000,165,2
     53,240,006,056,232,233,003
1260 DATA  208,251,169,001,133,2
     47,138,240,005,006,247,202
1270 DATA  208,251,024,165,247,1
     01,002,133,002,234,230,253
1280 DATA  230,253,230,253,165,2
     53,201,019,144,163,165,002
1290 DATA  032,210,255,024,144,0
     04,208,137,208,139,198,252
1300 DATA  016,139,230,251,165,2
     51,201,003,208,242,169,013
1310 DATA  032,210,255,230,250,1
     65,250,201,003,208,227,169
1320 DATA  015,032,210,255,032,2
     04,255,169,187,076,195,255,
     -1,40975
```

Disassemblato di "Stampa Sprite"
By G. Mariani
(Qui rilocato da $C000)

```
c000 lda #$00     ;Default per il banco
c002 sta $f7      ;(0).
c004 lda #$04     ;Default per la
c006 sta $f8      ;stampante (4).
c008 jsr $aefd    ;Controlla l' virgola
c00b jsr $b79e    ;e legge numero spr.
c00e lda #$00     ;Moltiplica il
c010 sta $fa      ;numero dello sprite
c012 stx $f9      ;per 64 e lo salva
c014 ldy #$06     ;in $F9/$FA.
c016 asl $f9      ;
c018 rol $fa      ;
c01a dey          ;
c01b bne $c016    ;
c01d lda #$2c     ;Controlla la presenza
c01f ldy #$00     ;della seconda virgola
c021 cmp ($7a),y  ;Se non c'e'...
c023 bne $c044    ;passa avanti.
c025 jsr $aefd    ;Altrimenti la salta
c028 jsr $b79e    ;Legge il banco.
c02b cpx #$04     ;Controlla se e' >3.
c02d bcc $c032    ;Se no, salta avanti.
c02f jmp $b248    ;ILLEGAL QUANTITY ERR.
c032 stx $f7      ;Salva il banco in $F7
c034 lda #$2c     ;Controlla la presenza
c036 ldy #$00     ;della terza virgola.
c038 cmp ($7a),y  ;Se non c'e'...
c03a bne $c044    ;passa avanti.
c03c jsr $aefd    ;Altrimenti la salta
c03f jsr $b79e    ;Legge il n. device.
c042 stx $f8      ;Lo salva il $F8.
c044 lda $f9      ;
c046 sta $fe      ;
c048 lda $f7      ;Moltiplica il numero
c04a asl          ;del banco x 16384.
c04b asl          ;
c04c asl          ;
c04d asl          ;
c04e asl          ;
c04f asl          ;Lo addiziona al
c050 cld          ;numero sprite x 64 e
c051 clc          ;lo salva il $FE/$FF
c052 adc $fa      ;(Chiamiamolo P=
c054 sta $ff      ;indirizzo iniziale)
c056 lda #$bb     ;
c058 ldx $f8      ;Apre il canale n.187
c05a ldy #$00     ;di comunicazione con
c05c jsr $ffba    ;la stampante.
c05f lda #$00     ;
c061 jsr $ffbd    ;
c064 jsr $ffc0    ;
c067 ldx #$bb     ;
c069 jsr $ffc9    ;
c06c lda #$08     ;Invia un CHR$(8) alla
c06e jsr $ffd2    ;stampante.
c071 lda #$00     ;Inizializzazione dei
c073 sta $fa      ;contatori per i vari
c075 lda #$00     ;cicli.(K=$FA,
c077 sta $fb      ;L=$FB, B=$FC, Q=$FD)
```

```
c079 lda #$07      ;
c07b sta $fc       ;
c07d lda #$80      ;Inizializza carattere
c07f sta $02       ;da stampare (V).
c081 lda #$00      ;
c083 sta $fd       ;
c085 lda $fa       ;Calcola indirizzo X
c087 clc           ;per prelevare il dato
c088 ldx #$14      ;secondo la formula:
c08a adc $fa       ;X=P+(Q+L+K*21).
c08c dex           ;Esegue r1=K*21
c08d bne $c08a     ;
c08f adc $fb       ;Somma r2=r1+L.
c091 adc $fd       ;Somma r3=r2+Q.
c093 ldx $ff       ;
c095 clc           ;Esegue la somma:
c096 adc $fe       ;X=P+r3.
c098 sta $f8       ;
c09a bcc $c09d     ;
c09c inx           ;
c09d stx $f9       ;
c09f ldy #$00      ;Esegue l'operazione
c0a1 lda #$01      ;r1=(2↑B)
c0a3 sta $f7       ;
c0a5 ldx $fc       ;
c0a7 beq $c0ae     ;
c0a9 asl $f7       ;
c0ab dex           ;
c0ac bne $c0a9     ;
c0ae lda ($f8),y   ;Carica: r2=PEEK(X)
c0b0 and $f7       ;Esegue r3=r1 AND r2.
c0b2 cmp $f7       ;Controlla: se r1<>r3
c0b4 bne $c0d6     ;(pix. spento), salta
c0b6 ldx #$00      ;Altrimenti esegue:
c0b8 lda $fd       ;V=V+2↑(Q/3).
c0ba beq $c0c2     ;
c0bc sec           ;Esegue divisione:
c0bd inx           ;r1=Q/3.
c0be sbc #$03      ;
c0c0 bne $c0bd     ;
```

```
c0c2 lda #$01      ;Esegue operazione:
c0c4 sta $f7       ;r2=2↑r1.
c0c6 txa           ;
c0c7 beq $c0ce     ;
c0c9 asl $f7       ;
c0cb dex           ;
c0cc bne $c0c9     ;
c0ce clc           ;Addiziona:
c0cf lda $f7       ;V=V+r2.
c0d1 adc $02       ;
c0d3 sta $02       ;
c0d5 nop           ;
c0d6 inc $fd       ;Incrementa il contat.
c0d8 inc $fd       ;Q di 3 volte (Q=Q+3)
c0da inc $fd       ;
c0dc lda $fd       ;Controlla: Q>18?
c0de cmp #$13      ;
c0e0 bcc $c085     ;No: Salta
c0e2 lda $02       ;Si: Stampa il
c0e4 jsr $ffd2     ;carattere.
c0e7 clc           ;Salta avanti.
c0e8 bcc $c0ee     ;
c0ea bne $c075     ;Salti indietro (rif.
c0ec bne $c079     ;piu' avanti)
c0ee dec $fc       ;Decrementa B.
c0f0 bpl $c07d     ;Se B<0, salta.
c0f2 inc $fb       ;Altrimenti increm. L
c0f4 lda $fb       ;Controlla:
c0f6 cmp #$03      ;L>2?
c0f8 bne $c0ec     ;Si: salta (rif.C079)
c0fa lda #$0d      ;No: Invia un CR alla
c0fc jsr $ffd2     ;stampante (nuova lin)
c0ff inc $fa       ;Incrementa il contat.
c101 lda $fa       ;K. Controlla:
c103 cmp #$03      ;K>2?
c105 bne $c0ea     ;Si: salta (rif.C075)
c107 lda #$0f      ;No: Invia un CHR$(15)
c109 jsr $ffd2     ;alla stampante.
c10c jsr $ffcc     ;Reset canali I/O.
c10f lda #$bb      ;Chiude il canale 187
c111 jmp $ffc3     ;e torna al Basic.
```

INTRODUZIONE ALLA PROSPETTIVA

Lo studio della prospettiva è sicuramente uno più difficili argomenti di grafica. E' possibile renderlo semplice per il nostro computer?

La rappresentazione tridimensionale degli oggetti affascina l'uomo da tempo immemorabile perchè, se ben realizzata, riesce ad esplicitare un numero incredibile di informazioni. Si sa, infatti, che una fotografia (che rappresenta l'optimum della tecnica tridimensionale) vale più di mille parole.

Una foto realizzata con un grandangolo "spinto", conferisce, ad esempio, un senso di notevole profondità.

Se, però, cerchiamo di scoprire la regola generale che classifica un disegno come tridimensionale, difficilmente riusciamo a scoprirne le leggi grafiche che, al contrario, sóno piuttosto semplici da applicare, almeno nei casi più comuni.

Non dobbiamo dimenticare, infatti, che...

"Costruire l'immagine prospettica di un oggetto vuol dire costruire una figura geometricamente simile a quella che l'oggetto imprime sulla retina dei nostri occhi".

Non ha molta importanza, quindi, che il disegno rispetti rigorosamente le regole della prospettiva: l'im-

SCHEDA TECNICA

Software per applicazioni grafiche

Hardware richiesto: computer di standard Ms-Dos (+ Gw-Basic), C/64, C/16, Plus/4, C/128, ma facilmente adattabile ad altri computer Commodore

Richiede (per il C/64) il simulatore di Ms-Dos e GW-Basic

Consigliato a tutti gli appassionàti di grafica

Anche i programmi pubblicati in queste pagine sono contenuti nel disco "Directory" di questo mese.

portante è l'impressione che il disegno riesce a suscitare.

IN PROSPETTIVA

La costruzione semplificata (cioè "ad effetto") di una immagine in prospettiva è straordinariamente semplice sia nel caso della prospettiva normale che in quella cosiddetta "centrale".

Nel primo caso (figura 1) è necessario tracciare, anzitutto, una linea orizzontale delimitata dai due punti estremi A ed A' (punti di fuga). In seguito, ipotizzando di disegnare un parallelepipedo, tracceremo il suo lato verticale (B-B') in modo che sia perpendicolare alla linea prima tracciata (detta linea di terra). Allineando con un righello il punto B con A' e, in seguito, ancora B con A, tracceremo due segmenti, di lunghezza opportuna, che delimiteranno le facce laterali del nostro solido. Allo stesso modo ci comporteremo per la parte inferiore, tracciando i segmenti A-B' e B'-A' tenendo presente che, in ogni caso, i lati verticali devono sempre essere paralleli tra loro e, di conseguenza, sempre ortogonali alla linea di terra.

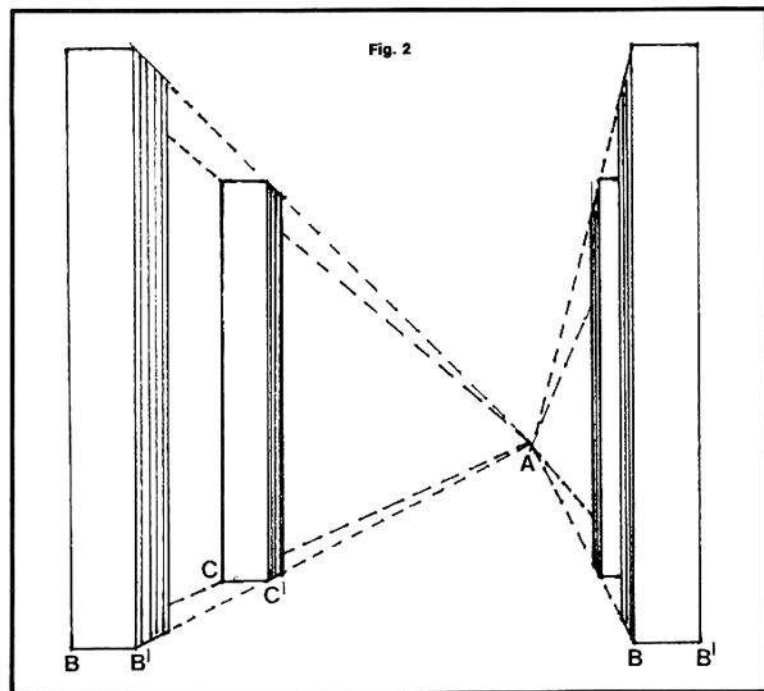
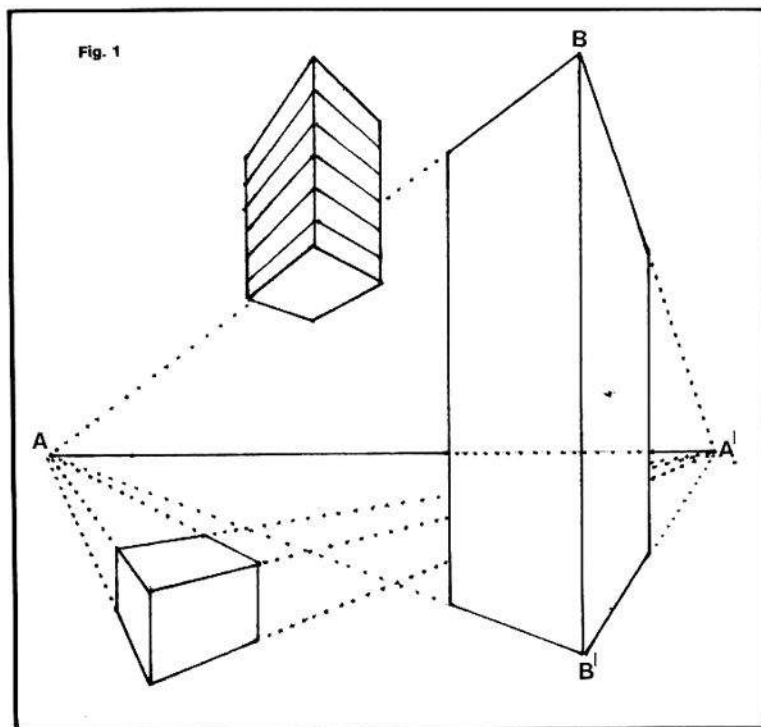



Fig. 1

Fig. 2

Naturalmente non entriamo nei dettagli sul come rendere proporzionali i fianchi del nostro solido: abbiamo solo voluto accennare alle regole generali che consentono di disegnare solidi in prospettiva. Si può notare facilmente che qualunque linea presente nella figura, con la sola eccezione delle linee verticali, incontra il punto A oppure il punto A': questa semplice regola empirica permette a chiunque di disegnare facilmente in prospettiva.

CENTRALIZZANDO

Il caso della prospettiva centrale (figura 2) si ha in alcuni casi particolari, difficili da esaminare in dettaglio, che si verificano, di solito, quando ci si trova in ambienti interni: se ci

Fig. 3

disponiamo al centro di una stanza, lungo le cui pareti sono presenti dei pilastri, ci troviamo in un caso di prospettiva centrale, come pure guardando all'interno di un pozzo.

In questo caso i due punti di fuga del disegno precedente (A ed A') sono presenti, ma all'infinito; ne consegue che, ad eccezione delle linee verticali, tutte le linee di un disegno rappresentato in questo tipo di prospettiva (oppure i loro prolungamenti) devono incontrare sempre un punto (indicato con "A" in figura 2).

Anche in questo caso uno studio approfondito sulla proporzionalità delle distanze in prospettiva esula dallo scopo delle presenti note.

Confrontando la "struttura" dello schema di figura 2 con i due triangoli di figura 3 possiamo notare una certa similitudine: il punto H (fig. 3) ha le stesse funzioni del punto A (di fig. 2) e le linee che partono dai vertici dei due triangoli sembrano parenti strette delle linee tratteggiate della stessa figura 2.

In effetti è proprio così: la larghezza delle colonne è compresa tra B-B' e C-C', vale a dire delimitata da linee orizzontali che intersecano le linee A-B ed A-B'.

Nella prospettiva centrale vale quindi una considerazione che deriva direttamente dal famoso teorema di Talete, secondo cui...

"Se un fascio di rette parallele (in particolare: A-D e E-L di fig. 3) è tagliato da rette trasversali (H-E, H-F, H-G, H-L che hanno, tra l'altro, la particolarità di avere il punto H in comune tra loro) i segmenti determinati su una trasversale (A-B, B-C, C-D) sono proporzionali ai segmenti corrispondenti sull'altra (E-F, F-G, G-L)"

Questa semplice regola ci permetterà, in seguito, di semplificare notevolmente il lavoro di programmazione perchè se è vero che le rette di una prospettiva centrale generano segmenti proporzionali tra loro, sarà anche vero che segmenti proporzionali tra loro possono essere considerati appartenenti ad un sistema prospettico centrale.

Dando uno sguardo ai numerosi disegni presenti in queste pagine, è possibile accorgersi facilmente che gli schemi "grezzi" ivi rappresentati rendono piuttosto bene l'idea della struttura base di una prospettiva centrale: il rettangolo visibile, infatti, è il muro, posto in lontananza, dell'ipotetica stanza all'interno della quale ci troviamo; le linee variamente inclinate simboleggiano, invece, le linee prospettiche già incontrate nella figura 2.

Nonostante le varie figure pubblicate "rendano" abbastanza bene l'idea, dobbiamo notare che è valida, geometricamente, solo la figura in cui le linee di fuga hanno tutte un punto in comune (come in figura 4). Le altre, tra cui la figura 5, non hanno tutte un punto in comune; le linee

Fig. 4

superiori condividono A e quelle inferiori, invece, B. Nonostante ciò l'occhio sembra non accorgersi dell'errore che può, comunque, portare a vistose contraddizioni nel caso si decida di utilizzare dette strutture per "ambientarvi" solidi di vario tipo.

I PROGRAMMI

I due programmi che compaiono in queste pagine sono scritti in due linguaggi: il Basic 7.0 del C/128 (compatibile con il Basic 3.5 del C/16 e Plus/4) ed il Gw-Basic del C/64. Chi possiede altri linguaggi interpreti sarà sicuramente in grado di apportare le necessarie (e semplicissime) modifiche del caso.

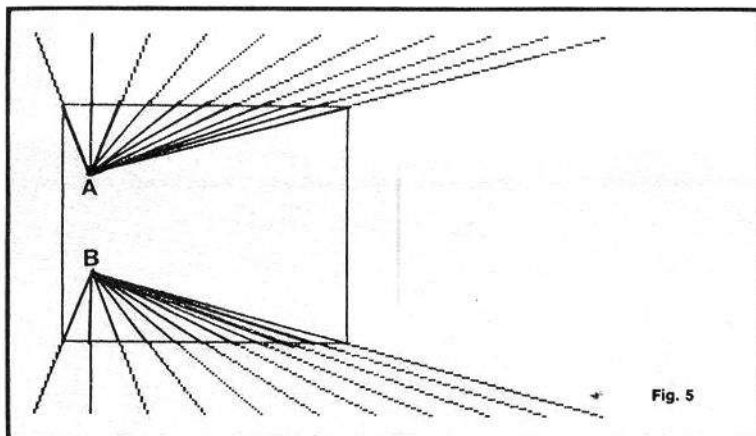



Fig. 5

```
100 REM PRIMI STUDI DI PROSPETTIVA
110 REM COMPUTER C/64 + GW-BASIC
120 REM O QUALSIASI MS-DOS COMPATIBILE
130 CLS 0
140 L1=100:INPUT"LUNGHEZZA 1";L1
150 L2=200:INPUT"LUNGHEZZA 2";L2
160 SP=30:  INPUT"SPIN (1-50)";SP
170 VS=L1/10:REM STEP MINORE
180 ES=L2/10:REM STEP MAGGIORE
190 V1=20:INPUT"VICIN.SIN.ALTO=";V1
200 V2=10:INPUT"VICIN.SIN.BASS=";V2
210 SCREEN 1:CLS 1
220 LINE (0,0)    -(319,0)
221 LINE (319,0)  -(319,199)
222 LINE (319,199)-(0,199)
230 LINE (0,199)  -(0,0)
240 LINE (V1,50)-(V1,150)
250 LINE (V1,150)- (V1+L1,150)
260 LINE (V1,50)-(V1+L1,50)
270 X=0:FOR I=0 TO L1 STEP VS
280 LINE (I+V1,150)- (V2+ES*X,150+SP)
290 LINE (I+V1,50)- (V2+ES*X,50-SP)
300 X=X+1:NEXT
310 LINE(I+V1-VS,150)-(I+V1-VS,50)
320 GET A$:IF A$="" THEN 320
325 REM ATTENZIONE AI CODICI ASCII
326 REM PER I "VERI" COMPUTER MS-DOS!
330 IF ASC(A$)=141THEN SCREEN 0:GOTO140
340 IF ASC(A$)=157 THEN V1=V1-30
350 X=1:V1=V1+15 :GOTO 210
```

Non appena si digita Run, compare una domanda relativa alla lunghezza del segmento che, tanto per semplificare, rappresenterà la larghezza della "stanza" che, in seguito verrà tracciata (segmento L1 di fig. 6).

L'altezza della stanza è fissa, dal momento che è delimitata dalle due ordinate 50 e 150 (vedi, soprattutto, le righe 280 e 290). Inutile dire che il lettore può apportare le modifiche desiderate tenendo conto che con i valori 50 e 150 si è voluta ottenere una figura simmetrica rispetto all'orizzontale del video (altri valori per la simmetria: 25-175, 80-120 ed altri tali che la loro somma sia 200).

La seconda domanda ("Lunghezza 2?") è relativa alla lunghezza del segmento (L2 di fig. 6) da prendere come riferimento per suddividerlo in parti proporzionali a L1.

In pratica i due segmenti, L1 ed L2, verranno divisi in 10 parti eguali ciascuno (righe 170-180) per creare, artificiosamente, la proporzionalità di cui abbiamo già parlato; in seguito sarà sufficiente congiungere le parti proporzionali per ottenere l'effetto prospettico!

La domanda relativa allo Spin (riga 160) consentirà di "regolare" l'inclinazione degli stessi segmenti: se, infatti, L1 ed L2 rimangono eguali ma

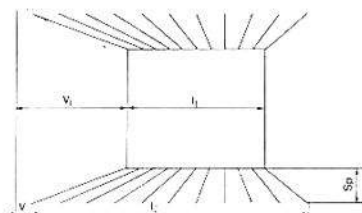



Fig. 6

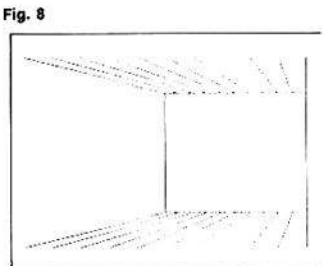



Fig. 7

Fig. 8

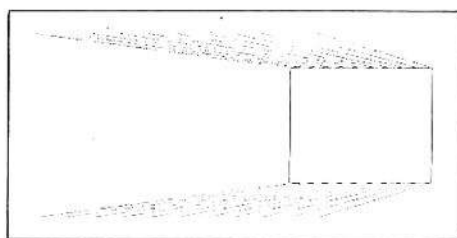
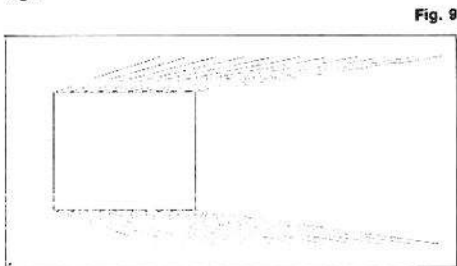



Fig. 9

si varia, appunto, la loro distanza, l'inclinazione dei segmenti prospettici varierà di conseguenza, conferendo, a seconda dei casi, l'aspetto di figura 4 oppure 5.

La domanda di riga 190 tende a stabilire la distanza in cui dovrà trovarsi lo spigolo sinistro del muro mentre la riga 200 assegnerà alla variabile V2 la distanza, sempre dal bordo sinistro del video, del segmento prospettico inferiore.

E' facile immaginare che, variando opportunamente i parametri di Input, la figura che verrà tracciata cambierà totalmente aspetto, simulando un'infinità di punti di osservazione, tra cui quello rigorosamente centrale, quello coincidente con uno dei muri laterali (figura 7) e, addirittura, posto al di fuori (destra o sinistra) degli stessi muri laterali (figure 8 e 9).

Se, durante la visualizzazione, premete un tasto qualunque, il punto di osservazione si sposterà a destra di 15 pixel (riga 350); premendo shift + Crsr si sposterà, invece, di 15 pixel a sinistra (riga 340). Con Shift + Return compariranno nuovamente le domande iniziali, allo scopo di impostare una nuova elaborazione.

Benchè superfluo, notiamo esplicitamente che la parte superiore della figura viene tracciata in modo automatico applicando una sola riga Basic che impone la simmetria (riga 290).

LE MODIFICHE

Grazie all'estrema semplicità del listato, consigliamo ai lettori di apportare alcune modifiche, in modo da realizzare "griglie" prospettiche più aderenti alla realtà. In particolare ci permettiamo di suggerire:

- Modifica, da Input, dell'altezza del muro frontale e suo posizionamento anche non simmetrico rispetto all'asse centrale orizzontale.
- Realizzazione di un fascio di rette orizzontali, parallele tra loro, che riproducano con efficacia l'effetto "mattonelle".
- Realizzazione della griglia prospettica anche lungo i "muri" laterali.

Ai più bravi, infine, suggeriamo di realizzare una procedura automatica in grado di riprodurre un grigliato che simuli l'mpostazione prospettica indicata in figura 1.

```
100 REM PRIMI STUDI DI PROSPETTIVA
110 REM COMPUTER C/128, C/16, PLUS/4
120 :
130 PRINT CHR$(147)
140 L1=100:INPUT"SLUNGHEZZA 1";L1
150 L2=200:INPUT"LUNGHEZZA 2";L2
160 SP=30:  INPUT"SPIN (1-50)";SP
170 VS=L1/10:REM STEP MINORE
180 ES=L2/10:REM STEP MAGGIORE
190 V1=20:INPUT"VICIN.SIN.ALTO=";V1
200 V2=10:INPUT"VICIN.SIN.BASS=";V2
210 GRAPHIC 1,1
220 DRAW 1,0,0 TO 319,0 TO 319,199
230 DRAW TO 0,199 TO 0,0
240 DRAW 1,V1,50 TO V1,150
250 DRAW 1,V1,150 TO V1+L1,150
260 DRAW 1,V1,50 TO V1+L1,50
270 X=0:FOR I=0 TO L1 STEP VS
280 DRAW 1,I+V1,150 TO V2+ES*X,150+SP
290 DRAW 1,I+V1, 50 TO V2+ES*X,50-SP
300 X=X+1:NEXT
310 DRAW 1,I+V1-VS,150 TO I+V1-VS,50
320 GET A$:IF A$="" THEN 320
330 IF ASC(A$)=141THEN GRAPHIC0:GOTO 140
340 IF ASC(A$)=157THEN V1=V1-30
350 X=1:V1=V1+15 :GOTO 210

READY.
```

commodore

COMPUTER CLUB

La rivista degli utenti di sistemi Commodore

games

...TRAZ

TRANSFORMABLE ARCADE ZONE

Break Out...into the New Generation.

Stealth Mission
The Chernobil Syndrome
Cybernoid
Blood Valley
Strip Poker
Jump Jet
Asterix
Hayes Invaders
The Wall
The Moty
Indoor Sports
Sidewinder
Tetris
Rolling Thunder
Time Bandit
Eco

COMMODORE 64 DISK

Come leggere le recensioni

Ogni mese, su queste pagine, verranno esaminati e testati i videogame più recenti per i computer Commodore 64 ed Amiga.

Ad ogni descrizione verranno associate una immagine, catturata tra le più belle schermate, ed una breve pagella.

Quest'ultima, pur se, inevitabilmente, frutto di impressioni personali di chi esamina il gioco stesso, ha lo scopo di assegnare una valutazione del livello del software, soprattutto tenendo conto di altri game analoghi disponibili sul mercato.

La pagella, comprende cinque voci:

IMPATTO: indica il livello di interesse suscitato dalla presentazione e dal tema del gioco.

SCENARIO: riguarda l'accuratezza con cui è realizzata la grafica e l'efficacia dei disegni degli sprite.

SUONO: valuta gli effetti sonori presenti e le eventuali musiche di sottofondo.

INTERESSE: si riferisce al livello di interesse che il gioco può suscitare in un giocatore abituato ai videogame e, implicitamente, alla sua probabilità di "permanenza" sui vostri monitor.

TOTALE: ha lo scopo di sintetizzare i precedenti valori con un voto unico.

STEALTH MISSION C/64-128

Sulla scia dei ben noti Flight Simulator I e II, ecco una nuova simulazione di volo dalla Sublogic.

Un gioco davvero avvincente, sia per l'aspetto grafico, sia per la completezza ed il realismo della simulazione.

E' possibile volare con un F-19, un X-29 oppure con l'F-14, di giorno o in missione notturna, volando sul mare o sulla terraferma.

Se ciò non bastasse, la vasta gamma di visuali (frontale, posteriore, dall'alto, laterale, dal satellite, dalla torre di controllo e da un ricognitore radiocomandato) contribuisce a rendere più appetibile questo gioco, anche per coloro che posseggono migliaia di videogame.

Durante il volo sarà di aiuto il pannello informativo, posto nella parte bassa dello schermo.

Anche se puoi armare l'aereo come meglio credi, in relazione al tipo di nemico da affrontare, in casi estremi puoi sempre salvarti con il sedile ad espulsione...

IMPATTO:	*8*
SCENARIO:	*6*
SUONO:	*6*
INTERESSE:	*8*
TOTALE:	**8**

THE CHERNOBYL SYNDROME C/64-128

L'incidente della centrale nucleare di Chernobyl, che all'epoca destò l'interesse di tutti i mass media, sembra essere caduto nell'oblio; ma a quanto pare qualcosa è rimasto nell'aria...

Non spaventatevi: stiamo solo parlando di un videogame per C/64; "The Cernobyl syndrome" è infatti ambientato all'interno della tristemente famosa centrale nucleare.

Si tratta di una simulazione, nella quale il vostro ruolo è quello di controllare il perfetto funzionamento di tutti gli impianti, operando su un sistema di sicurezza computerizzato.

Nel caso di avarie dovrete rispondere tempestivamente con la soluzione più congeniale, e se questa dovesse rivelarsi inefficace, non rimarrà che evacuare la centrale, prima che sia troppo tardi!

IMPATTO:	*7*
SCENARIO:	*6*
SUONO:	*6*
INTERESSE:	*5*
TOTALE:	**6**

CYBERNOID C/64-128

Chi ama i giochi di azione, che non lasciano neanche il tempo di respirare, in "Cybernoid" trova pane per i suoi denti.

Il gioco, oltre ad offrire lo stimolo della rapidità d'azione, è corredato da stupende schermate, ed altrettanto efficaci effetti sonori.

E' possibile utilizzare la tastiera od il joystick, ed attivare gli effetti sonori in alternativa alla colonna musicale.

La trama del gioco consiste nel recuperare munizioni, metalli preziosi e minerali per la produzione di energia, che sono stati rubati dai depositi della federazione interstellare.

L'azione di gioco si svolge all'interno di diversi quadri a scomparsa, per ognuno dei quali è stabilito un tempo limite, pena la distruzione di un Cybernoid.

IMPATTO:	*8*
SCENARIO:	*9*
SUONO:	*8*
INTERESSE:	*8*
TOTALE:	**8**

BLOOD VALLEY C/64-128

In un pianeta recentemente scoperto, ma con vaste zone ancora inesplorate, in una valle dallo squallido paessaggio lunare, si consumano periodicamente rituali crudeli.

Il sovrano della "Blood valley", per puro divertimento, di tanto in tanto indice la caccia allo schiavo.

Questa crudele farsa avrà termine solo quando il sovrano si sarà stancato, con l'uccisione dello schiavo.

E' prevista l'opzione per uno o due giocatori.

IMPATTO:	*6*
SCENARIO:	*6*
SUONO:	*5*
INTERESSE:	*5*
TOTALE:	**5**

STRIP POKER II
Amiga

La riedizione del gioco originariamente scritto anche per Apple e PET/CBM. La prima versione aveva le donnine che si spogliano quando perdono le scommesse in un pessimo IFF, ora non si è ancora allo HAM (come in Hollywood Poker, già recensito), ma le cose vanno meglio, anche per quanto riguarda i controlli (via mouse) e la grafica delle carte e delle opzioni.

IMPATTO:	*6*
SCENARIO:	*7*
SUONO:	*5*
INTERESSE:	*5*
TOTALE:	**5**

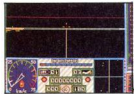
JUMP JET
Amiga

Si tratta di una bella simulazione di stampo videogiocoso, non "Professionale", in cui campeggia un elicottero a decollo verticale. La grafica è ottima e l'ambientazione decente. Metà schermo è occupato dal quadro dei controlli (per leggere tutto è indispensabile il manuale), mentre l'altra metà presenta la scena vera e propria, con diversi modi possibili di visione.

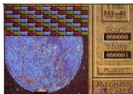
IMPATTO:	*8*
SCENARIO:	*8*
SUONO:	*7*
INTERESSE:	
TOTALE:	**8**

ASTERIX
Amiga

Il difetto essenziale di questo gioco è che, scritto in Germania, è in lingua tedesca. La quale cosa, per un programma dotato di grafica del famosissimo personaggio di Goscinny e Uderzo (assolutamente fedele, con resa grafica impressionante e non concepibile su macchine inferiori ad Amiga), che parla spesso con fumetti, fa imbestialire. Già abbiamo preso ripetizioni per imparare l'inglese e potere videogiocare, ora ci chiedono anche la conoscenza del tedesco! Comunque, nessun programma è mai stato così vicino, come questo, ad un vero cartone animato.

HAYES INVADERS
Amiga

Uno strano gioco: è un "invasori", dotato di una grafica che, colori e nitidezza a parte, si poteva forse fare meglio sul C/64 in Basic. Lo scopo è di sparacchiare a tutto ciò che si muove, o quasi, ma con effetti davvero mediocri. La grafica è pseudo 3D, dotata di uno scorrimento vergognoso. Una versione migliore si poteva effettuare sul Vic/20 (privo di espansione).

IMPATTO:	*4*
SCENARIO:	*3*
SUONO:	*2*
INTERESSE:	*2*
TOTALE:	**2**

THE WALL
Amiga

Ennesima versione di Arkanoid (dopo l'originale, Crystal Hammer, Impact, Amegas...). Qui la novità è che la pallina ha dei rimbalzi più cattivi che nell'originale e la grafica di sottofondo consiste di immagini IFF comprendenti anche belle ragazze piuttosto discinte. I mattoncini, abbattuti dalla pallina fatta rimbalzare dalla racchetta, scoprono così, a volte, bei paesaggi di vari tipi (con montagne verdi e montagnette rosa...).

IMPATTO:	*6*
SCENARIO:	*7*
SUONO:	*7*
INTERESSE:	*7*
TOTALE:	**6**

THE MOTY
Amiga

Uno di quei giochi che consistono nell'evitare, più che nel colpire. Un omino si aggira per vari schermi spostando pietre ed evitando oggetti vari. A parte la banalità, la grafica sarebbe stata ridicola addirittura in programmi del C/64 di tre anni fa; la musica è paragonabile a quella dello Spectrum (che non ha il SID, ma solo un cicalino piezoelettrico). Forse può essere utile per "iniziare" il vostro fratellino, molto piccolo, all'Amiga.

IMPATTO:	*4*
SCENARIO:	*4*
SUONO:	*2*
INTERESSE:	*3*
TOTALE:	**4**

INDOOR SPORTS
Amiga

E', in realtà, una compilation di quattro giochini: ping-pong, freccette, bowling ed una specie di calcetto su ghiaccio a due. Tutti i giochi hanno una grafica simpatica e abbastanza rifinita, come solo Amiga consente. Ciò che manca è un tocco di originalità e gli incentivi nel gioco. E' possibile comunque confrontarsi sino a quattro per volta a suon di joystick. Solo per i patiti dei giochi banalotti.

IMPATTO:	*7*
SCENARIO:	*7*
SUONO:	*6*
INTERESSE:	*4*
TOTALE:	**5**

SIDEWINDER
Amiga

Un altro gioco stile URIDIUM, cioè con una astronave che si muove su piattaforme e pianeti che scorrono dall'alto verso il basso. Si deve sparare a tutto ciò che si muove e che non si muove. E' letale rimanere senza carburante o cozzare contro astronavi o protuberanze della base o del suolo. Si può giocare anche in due. La grafica è un po' lenta (specie se paragonata all'incredibile Insanity Fight svizzero), ma più piccola e, quindi, più visibile.

IMPATTO:	*7*
SCENARIO:	*7*
SUONO:	*6*
INTERESSE:	*5*
TOTALE:	**6**

TETRIS
Amiga

E' uno di quei videogiochi che, apparentemente semplicissimi, richiedono molte ore di meditazione per essere compresi, se non si leggono (o se non si possiedono...) le istruzioni. Infatti bisogna comporre spezzoni di figure geometriche che cadono dall'alto in un modo intelligente, per realizzare costruzioni accettabili. Si tratta di una specie di puzzle animato, con colori vivaci, ma grafica lentissima.

IMPATTO:	6
SCENARIO:	5
SUONO:	6
INTERESSE:	6
TOTALE:	**5**

ROLLING THUNDER
Amiga

Un bel gioco ispirato al famoso "Buck Roger" (quello con l'omino che spara, non con l'astronave). Ci si muove in una base aliena, con nemici di tutti i tipi che ci vogliono uccidere. Dobbiamo visitare luoghi, usare il laser, approvvigionarci di proiettili, saltare balaustre ed, infine, eliminare tutti quei mostriciattoli verdi. Il gioco è del tipo "sarebbe perfetto", se solo la grafica fosse stata più curata

IMPATTO:	8
SCENARIO:	7
SUONO:	8
INTERESSE:	8
TOTALE:	**8**

TIME BANDIT
Amiga

Dalla Microdeal un gioco con grafica in prospettiva quasi-3d "dall'alto", non molto convincente, occupante mezzo schermo. Lo scorrimento della finestra è continuo, ma non molto raffinato e la grafica è di piccole dimensioni, ma ricchissima di particolari. In pratica si tratta di un Pacman evolutissimo: il labirinto è molto più complesso, come tutto il resto, poi vi sono oggetti da collezionare e piccoli problemi da risolvere

IMPATTO:	7
SCENARIO:	7
SUONO:	6
INTERESSE:	6
TOTALE:	**6**

ECO
Amiga

Dalla rediviva Ocean, un originalissimo gioco. Tutte le forme di vita, secondo Darwin, si sono evolute da forme semplici. Il giocatore (ma si può giocare in due) è una specie di ragno che deve sopravvivere in un mondo primordiale, povero di cibo e ricco di predatori. La grafica è in prospettiva tridimensionale, mentre nella porzione inferiore dello schermo è presente un radar che indica dove sono i nemici e dove il cibo.

IMPATTO:	8
SCENARIO:	8
SUONO:	8
INTERESSE:	7
TOTALE:	**8**

QUANTO COSTA IL TUO COMMODORE

LISTINO PREZZI HOME COMPUTER

Codice	Prodotto	Prezzo (IVA esclusa)
	Personal Computer	
A 500	Personal Computer 16/32 BIT CPU 512K RAM incorpora un floppy disk drive da 3" 1/2 da 880Kb e un mouse	**L. 950.000**
A 501	Espansione di memoria per Amiga 500 512Kbytes e orologio	**L. 212.500**
A 520/1	Modulatore per Amiga 500 permette di collegare Amiga 500 ad un qualsiasi televisore b/n e colori	**L. 40.000**
A SCART	Cavo collegamento Amiga 500 con TV prescart	**L. 27.000**
Nuovo C64	Nuovo personal computer CPU 64K RAM computer ad alta risoluzione grafica, 256 combinazioni di colori, sintetizzatore di suono. Collegabile ad un qualsiasi televisore	**L. 375.000**
1764	Espansione di memoria per C64 256Kbytes	**L. 195.000**
C 128D	Personal computer CPU 128Kb RAM CPU 128Kbytes espandibile a 512Kbytes, 48Kbytes ROM, basic 7,0. Tastiera separata. Alta risoluzione grafica, 16 colori + 8 sprites. 40 80 (RGB) colonne. Programmabile in CP/M 3.0. Funzionante in modo C64. Floppy disk da 340Kb incorporato.	**L. 895.000**
1700	Espansione di memoria per C128. 128Kb	**L. 165.000**
1750	Espansione di memoria per C128. 512Kb	**L. 235.000**
	Registratori	
1530	Registratore compatibile C64 - C128 - C128D	**L. 55.000**
	Unità a dischi	
1541 II	Floppy disk drive 5" 1/4. Unità di memoria di massa, drive singolo, capacità 170Kbytes in linea. Compatibile con C64	**L. 395.000**
1581	Floppy disk drive 3" 1/2. Unità di memoria di massa, drive singolo da 3" 1/2, capacità 800Kbytes. Compatibile C64 - C128 - C128D	**L. 420.000**
A 1010	Floppy disk drive esterno 3" 1/2	**L. 395.000**
1802	Monitor a colori, alta risoluzione, 14" con audio. Collegabile al C64, C128, C128D (40 colonne)	
2080	Monitor a colori alta persistenza alta risoluzione, 14", con audio antiriflesso. Collegabile ad Amiga PC, C64, C128, C128D. 640x400 pixel, 4096 colori	**L. 630.000**
1084	Monitor a colori. Alta risoluzione, 14", con audio, antiriflesso. Collegabile al C64 - C128 - C128D - Amiga - PC. 640x400 pixel, 4096 colori.	**L. 545.000**
	Stampanti	
MPS 1200 S	Stampante. 80 colonne, 120 cps, bidirezionale, carta in modulo singolo e trascinamento modulo continuo per C64 - C128 - C128D	**L. 480.000**

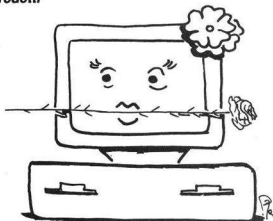
MPS 1200 P	Stampante 80 colonne 120 cps, bidirezionale, carta in modulo singolo e trascinamento modulo continuo per A500	**L. 480.000**
MPS 1250	Stampante con stesse caratteristiche del modello MPS 1200 P, interfacce seriale RS232C e parallela Centronics di serie	**L. 495.000**
MPS 1500	Stampante a colori. 80 colonne, 130 cps, 4 colori, bidirezionale, carta in modulo singolo e trascinamento modulo continuo.	**L. 550.000**
	Accessori	
6499	Adattatore telematico omologato. Collegabile al C64, permette il collegamento a Videotel PGE-Banche Dati.	**L. 149.000**
1351	Mouse. Per C64, C128, C128D	**L. 99.000**
1311	Joystick. Comando per giochi	**L. 13.500**
1399	Joystick. 8 microswitch-autofire	**L. 29.000**

LISTINO PREZZI SISTEMI

Codice		**Prezzo (IVA esclusa)**
PCI	Microprocessore 8088 (coprocessore 8087 opzionale) frequenza clock 4,77MHz. RAM 512Kb. Espandibile a 640Kb; 1 floppy da 360Kb. Monitor a fosfori verdi 12".	**L. 945.000**
PC 10-III	Microprocessore Intel 8088 a 16 bit; clock 10 MHz; memoria RAM: 640 Kb; memoria ROM (BIOS): 8 Kb autoconfigurabile; sistema operativo MS-DOS 3.21; GW-Basic; scheda video integrata monocromatico/colore di serie con emulazioni CGA/MDA/Hercules via software; 2 floppy disk drive da 360 Kb; monitor monocromatico a fosfori verdi da 12"; porte seriale RS232C e paralella Centronics; tastiera avanzata 102 tasti con 12 tasti funzione; porta mouse, incluso controller per H.D.	**L. 1.490.000**
PC 10-IIIC	Stessa configurazione ma con monitor 14" a colori mod. 1084	**L. 1.890.000**
PC 20-III	Microprocessore Intel 8088 a 16 bit; memoria RAM da 640Kb; memoria ROM (BIOS) da 8Kb autoconfigurabile. Sistema operativo MS-DOS 3.21. Scheda video monocromatico/colore di serie con emulazioni CGA/MDA/Hercules via software. Un hard disk da 20Mb e un floppy disk drive da 360Kb. Monitor monocromatico a fosfori verdi da 12". Porta seriale RS232C parallela Centronics. Porta mouse incluso controller per H.D.; tastiera avanzata 102 tasti con 12 tasti funzione	**L. 2.240.000**
PC 20-IIIC	Stessa configurazione ma con monitor 14" a colori mod. 1084	**L. 2.640.000**
PC 40/20 AT	Microprocessore 80286 a 16 bit; clock 10/6 MHz; memoria RAM dal 1 Mb; hard disk da 20 Mb; un floppy disk drive da 1,2 Mb; Sistema operativo MS-DOS 3.2, GW-Basic; scheda video monocromatico/colore 132 colonne AGA di serie; monitor monocromatico a fosfori verdi da 14"; porta seriale RS232C e parallela Centronics; tastiera avanzata 102 tasti con 12 tasti funzione	**L. 3.490.000**
PC 40/40 AT	Microprocessore 80286 a 16 bit; clock 10/6 MHz; memoria RAM da 1Mb. Un hard disk da 40Mb; un floppy disk drive da 1,2Mb. Sistema operativo MS-DOS3.2, GW-Basic. Scheda video monocromatico/colore EGA WONDER di serie. Monitor monocromatico a fosfori verdi da 14". Porta seriale RS232 e parallela Centronics; tastiera avanzata 102 tasti con 12 tasti funzione	**L. 4.490.000**
PC 60/40	Microprocessore 80386 a 32 bit; clock 8/16 MHz; memoria RAM da 2,5 MB; hard disk da 40 Mb; 1 floppy disk drive da 1,2 Mb; sistema operativo MS-DOS 3.20; scheda EGA WONDER di serie; monitor ADI monocromatico a fosfori verdi da 14"; 2 porte seriali RS232C e 2 parallele Centronics. In opzione: floppy disk drive da 3" 1/2 e 1.44 Mb. Tastiera avanzata 102 tasti con 12 tasti funzione	**L. 8.490.000**
PC 60/80	Stessa configurazione del PC 60/40 ma con hard disk da 80 Mb e, in più, MS-DOS WINDOWS 386 e mouse	**L. 9.990.000**

Amiga 2000	Microprocessore Motorola 68000; clock 7,16 MHz; 1MB RAM; 256KB ROM; kickstart in ROM. Uscita PAL; 4 uscite a 4 voci su 2 canali (stereofonia). Amiga DOS; Amiga multitasking. 7 slots di sistema (2 combinati Amiga - Ibm PC AT); 1 slot video; 1 accesso diretto CPU 86 pin. Compatibilità MS-DOS interna con schede Janus (XT/AT CARD). Possibilità di un secondo drive interno da 3" 1/2 e di un drive interno da 5" 1/4. Monitor a colori 1084 tastiera; mouse. Workbench; Extras. MONITOR ESCLUSO	**L. 2.005.000**
	Monitor	
2080	Monitor a colori ad alta risoluzione e lunga persistenza 14" con audio; 640x400 pixel; 4096 colori per A500/A1000/A2000/PC AT	**L. 630.000**
1084	Monitor a colori ad alta risoluzione 14" con audio; antiriflesso; 640x400 pixel; 4096 colori per A500/A1000/A2000/PC AT	**L. 545.000**
1402	Monitor monocromatico da 12"	**L. 255.000**
1450	Monitor monocromatico BISYNC	**L. • □**
1950	Monitor colore BISYNC	**L. • □**
A 2024	Monitor monocromatico alta risoluzione a pagina bianca	**L. • □**
	Stampanti	
MPS 1200P	Stampante a 9 aghi 120 cps bidirezionale 80 colonne. Interfaccia Commodore per PC 10/III PC 40 PC/60/80	**L. 480.000**
MPS 1250	Stesse caratteristiche del modello MPS 1200P, interfacce seriale RS232C e parallela Centronics di serie	**L. 495.000**
MPS 1500	Stampante a colori a 9 aghi 130 cps; bidirezionale; 80 colonne; carta in modulo singolo e trascinamento modulo continuo; 4 colori; per Amiga e PC	
	Unità dischi	
K 910	Floppy disk drive aggiuntivo da 3" 1/2 e 1.44 Mb per PC 10-III e PC 20-III	**L. 295.000**
A 1010	Floppy disk drive esterno da 3" 1/2	**L. 395.000**
A 2010	Floppy disk drive interno da 3" 1/2	**L. 290.000**
A 2092 + PC 5060	Hard disk da 20 Mb settorizzabile PC/Amiga + scheda controller per hard disk per A 2000	**L. 845.000**
A 2080 + A 2020	Scheda Janus XT compatibile + drive	**L. 895.000**
A 2286 + A 2020	Scheda Janus II AT compatibile + drive	**L. 1.285.000**
A 2620	Scheda 68020 con processore a 32 bit; 14,28 MHZ	**L. •**
A 2090 + A 2092	Hard disk 20 Mb; interfaccia SCSI per A 2000	**L. 1.025.000**
A 2090 + A 2094	Hard disk 40 Mb; interfaccia SCSI per A 2000	**L. 1.575.000**
A 590	Hard disk da 20 Mb + fast RAM per A 500	**L. • □**
	Accessori	
PC EXP1	Box esterno per espansione PCI; può contenere 3 schede aggiuntive e hard disk alimentazione e cavi compresi	**L. 590.000**
A 2052	Scheda di espansione RAM da 2 Mb per A 2000 (max 4)	**L. 590.000**
A 2060	Scheda video/RF molulatore per TV e VCR per A2000	**L. 138.000**
A 2058	Espansione di memoria da 8 Mb RAM per A2000	**L. 3.760.000**
A UNIX	Scheda UNIX utilizzabile con A2620 (68020 board)	**L. • □**
1352	Mouse due tasti collegabile alla serie dei PC include Microsoft Mouse Driver	**L. 72.000**
A 2300	Genlock card semiprofessionale per A2000; permette di miscelare grafici con videocamere VCR; selettore per video Amiga, monitors esterni, proiettori; include software per titolazione	**L. 345.000 Δ**
A 2350	Professional Video Adapter Card per A 2000. Genlock professionale con freeze frame e digitalizzatore video incluse software di controllo per la gestione interattiva	**L. 1.645.000 Δ**

• prezzo da definire
Δ disponibile da giugno '88
□ disponibile da settembre '88

I primi 100 negozi selezionati dalla "Commodore" per la vendita dei suoi prodotti

Al Risparmio	*V.le Monza 204*	20128 Milano	Tel. 02-2573440
Braha Alberto S.D.F.	*Via Pier Capponi 5*	20145 Milano	Tel. 02-437468
E.D.S.	*C.so Porta Ticinese 4*	20123 Milano	Tel. 02-8322045
ESC	*Via Roggia Scagna 7*	20127 Milano	Tel. 02-2871300
Faref	*Via A. Volta 21*	20121 Milano	Tel. 02-650042
Giglioni	*Via D'Ovidio 8*	20131 Milano	Tel. 02-2360397
Giglioni	*V.le Luigi Sturzo 45*	20154 Milano	Tel. 02-654906
Logitek	*Via Golgi 60*	20133 Milano	Tel. 02-538931
Marcucci	*Via F.lli Bronzetti 37*	20129 Milano	Tel. 02-7386051
Melchioni	*Via P. Colletta 37*	20135 Milano	Tel. 02-57941
Messaggerie Musicali	*Galleria Del Corso*	20100 Milano	Tel. 02-50841
Newel	*Via Mac Mahon 75*	20155 Milano	Tel. 02-323492
Rivola	*Via Vitruvio 43*	20124 Milano	Tel. 02-6694160
F.lli Galimberti	*Via Naz. dei Giovi 28/36*	20030 Barlassina (Mi)	Tel. 0362-560625
P. Giorgio Ostellari	*Via Molino Arese 65*	20031 Cesano Mad.(Mi)	Tel. 0362-504392
P. Giorgio Ostellari	*Via Milano 300*	20233 Desio (Mi)	Tel. 0362-621042
GBC Italiana	*V.le Matteotti 66*	20092 Cinisello B. (Mi)	Tel. 02-6181801/ 6189391
	Via Petrella 6	20100 Milano	Tel. 02-203607/8
	Via Cantoni 7	20100 Milano	Tel. 02-437478
Casa Della Musica	*Via Indipendenza 21*	20093 Cologno Mon.(Mi)	Tel. 02-2542117
Penati	*Via Verdi 28/30*	20011 Corbetta (Mi)	Tel. 02-9779401
EPM System	*V.le Italia 12*	20094 Corsico (Mi)	Tel. 02-4407979
Centro Comp. Pandolfi	*Via Corridoni 18*	20025 Legnano (Mi)	Tel. 0331-546426
Computeam	*Via Vecellio 41*	20035 Lissone (Mi)	Tel. 039-481010
Futura	*Via Solferino 31*	20075 Lodi (Mi)	Tel. 0371-54457
MBM Inf. Sys.	*Corso Roma 112*	20075 Lodi (Mi)	Tel. 0371-53610
L'Amico del computer	*V.le Lombardia 17*	20077 Melegnano (Mi)	Tel. 02-9838341
Bit 84	*Via Italia 4*	20052 Monza (Mi)	Tel. 039-320813
I.C.O.	*Via dei Tigli 14*	20090 Opera (Mi)	Tel. 02-5242146
Comif	*Via Autolinee 10*	24100 Bergamo	Tel. 035-218553
Cordani	*Via dei Caniana 8*	24100 Bergamo	Tel. 035-258184
D.R.B.	*Via Borgo Palazzo 65*	24100 Bergamo	Tel. 035-237292
New Systems	*Via Paglia 36*	24100 Bergamo	Tel. 035-248109
Computer Team hi-tec	*Via Verdi 1/B*	24030 Carvico (Bg)	Tel. 035-790244
Ott. Optometr. Rovetta	*P.zza Garibaldi 6*	24065 Lovere (Bg)	Tel. 035-960705
A.I.S. International	*Via San Carlo 25*	24016 S.Pellegrino T.(Bg)	Tel. 0345-22662
Sisthema	*Via Roma 45*	24067 Sarnico (Bg)	Tel. 035-910750
Computer Center	*Via Cipro 62*	25125 Brescia	Tel. 030-223230
Informatica 2000	*Via Stazione 16/B*	25100 Brescia	Tel. 030-54015
Vigasio Mario	*Portici Zanardelli 3*	25100 Brescia	Tel. 030-59330/ 295858
Mister Bit	*Via Mazzini 70*	25043 Breno (Bs)	Tel. 0364-22835
Cavalli Pietro	*Via 10 Giornate 14B*	25030 Castrezzato (Bs)	Tel. 030-714013
Vietti Giuseppe	*Via Milano 1/B*	25032 Chiari (BS)	Tel. 030-7100206
Megabyte	*P.zza Duomo 17*	25015 Desenzano G.(Bs)	Tel. 030-9144880
Ditta Resi Rino	*Via XX Settembre 7*	25016 Ghedi (BS)	Tel. 030-901224
Info Cam	*Provinciale 38*	25050 Gratacasolo (Bs)	Tel. 0364-89379
Il Computer	*Via Indipendenza 90*	22100 Como	Tel. 031-240959
2M Elettronica	*Via Sacco 3*	22100 Como	Tel. 031-278227
Eletrongros	*Via L. Da Vinci 54*	22062 Barzanò (Co)	Tel. 039-957318
Ega	*Via Mazzini 42*	22065 Cassago Br.(Co)	Tel. 039-956307
Ega	*Via Aldo Moro 17*	22043 Galbiate (Co)	Tel. 0341-522028
Data Found Comp. Sh.	*Via A. Volta 4*	22036 Erba (Co)	Tel. 031-645761
Righi Elettronica	*Via G. Leopardi 26*	22077 Olgiate Com. (Co)	Tel. 031-946766
Fumagalli	*Via Cairoli 48*	22053 Lecco (Co)	Tel. 0341-863341
Cima Eelettronica	*Via Leonardo Da Vinci 7*	22053 Lecco (Co)	Tel. 0341-371106
Mondo Computer	*Via Giuseppina 11/B*	26100 Cremona	Tel. 0372-882079
Prisma	*Via Buoso Da Dovara 8*	26100 Cremona	Tel. 0372-437900
Telco	*P.zza Marconi 2/A*	26100 Cremona	Tel. 0372-31544
Elcom/GBC	*Via IV Novembre 56/58*	26013 Crema (Cr)	Tel. 0373-83393
Euroelettronica	*Via XX Settembre*	26013 Crema (Cr)	Tel. 0373-86966
Kofbaker	*Via Marchi 65/B*	26039 Vescovato (CR)	Tel. 0372-818601
Computer	*Galleria Fermi 7*	46100 Mantova	Tel. 0376-325616
32 Bit (Comp. St.)	*Via Cesare Battisti 14*	46100 Mantova	Tel. 0376-326770
Elettronica di Basso	*V.le Risorgimento 69*	46100 Mantova	Tel. 0376-329311
Poliware	*C.so C. Alberto 76*	27100 Pavia	Tel. 0382-22636
Log. Infor. Com. Shop	*V.le Monte Grappa 32*	27029 Vigevano (Pv)	Tel. 0381-81883
M. Visentin	*C.so Vitt. Emanuele 76*	27029 Vigevano (Pv)	Tel. 0381-83833
Computer Line	*Via G. Carducci 4*	29100 Piacenza	Tel. 0523-30691
Delta Computer	*Via Mart. della Resistenza 15/4*	29100 Piacenza	Tel. 0523-753318
Sover	*Via IV Novembre 60*	29100 Piacenza	Tel. 0523-34388
Cipolla Mauro	*Via Tremogge 25*	23100 Sondrio	Tel. 0342-213569
Fotonova		23010 S. Pietro Ber. (So)	Tel. 0342-492319
Dimeco Sistemi	*Via Garibaldi*	21100 Varese	Tel. 0332-237201
Il Centro Elettronico	*Via Morazzone 2*	21100 Varese	Tel. 0332-231006
Supergames	*Via Carrobbio 13*	21100 Varese	Tel. 0332-241092
Busto Bit	*Via Gavinana 17*	21052 Busto Arsizio (Va)	Tel. 0331-625034
Crespi Giuseppe & C.	*V.le Lombardia 59*	21053 Castellanza (Va)	Tel. 0331-503023
Computer Shop	*Via A. Da Brescia 2*	21013 Gallarate (Va)	Tel. 0331-798612
Lima Import-Export	*Via Clerici 196*	21040 Gerenzano (Va)	Tel.
J.A.C. nuove tecn.	*Via Matteotti 38*	21018 Sesto Cal. (Va)	Tel. 0331-923134
Bit Micro	*Via Mazzini 102*	15100 Alessandria	Tel. 0131-443252
West Records	*C.so Roma 85*	15100 Alessandria	Tel. 0131-441090
S.G.E. Elettronica	*Via Bandello 19*	15057 Tortona (Al)	Tel. 0131-867709
Record di Forina G.	*C.so Alfieri 166/3*	14100 Asti	Tel. 0141-34240
Il Diagramma Italia	*P.le Liberta' 4*	12100 Cuneo	Tel. 0171-57067
Rossi Computers	*C.so Nizza 42*	12100 Cuneo	Tel. 0171-2339
Punto Bit	*C.so Langhe 26/C*	12051 Alba (Cn)	Tel. 0173-49833
SDI	*Via Vitt. Emanuele 250*	12042 BRA (CN)	Tel. 0172-421392
Aschieri Gianfranco	*C.so Eman. Filiberto 6*	12045 Fossano (CN)	Tel. 0172-62995
Curetti Augusto	*C.so Italia 3*	12054 Mondovì (Cn)	Tel. 0174-42014
Ditta Elettrogamma	*C.so Risorgimento 20*	28100 Novara	Tel. 0321-176358
Elcom	*C.so Mazzini 11*	28100 Novara	Tel. 0321-391293
Programma 3	*V.le Buonarroti 8*	28100 Novara	Tel. 0321-36367/ 399903
Punto Video	*C.so Risorgimento 391*	28100 Novara	Tel. 0321-477367
Computer	*Via Monte Zeda 4*	28041 Arona (No)	Tel.
All Computer	*C.so Garibaldi 106*	28021 Borgomanero (No)	Tel. 0322-844142
Micrologic	*Via Giovanni XIII 2*	28037 Domodossola (No)	Tel.
Elliott Computer Shop	*Via Don Minzoni 32*	28044 Intra (No)	Tel. 0323-43517
ABA Elettronica	*Via C. Fossati 5/P*	10100 Torino	Tel. 011-302065
Alex Computer e giochi	*C.so Francia 333/4*	10142 Torino	Tel. 011-7730184
C.D.M. Elettronica	*Via Marochetti 17*	10126 Torino	Tel. 011-636345/ 634900
Computing New	*Via Marco Polo 40/E*	10129 Torino	Tel. 011-501512
De Bug	*C.so Vitt. Emanuele II 22*	10100 Torino	Tel. 011-832986
Desme Universal	*Via San Secondo 95*	10100 Torino	Tel. 011-592551
F.D.S.	*Via Borgaro 86/D*	10100 Torino	Tel. 011-2168900
Computer Home	*Via San Donato 46/D*	10100 Torino	Tel. 011-4731196
Informatica Italia	*C.so Re Umberto 129*	10128 Torino	Tel. 011-501647
MT Informatica	*C.so Giulio Cesare 58*	10100 Torino	Tel. 011-850955/ 238803
New Bus. Computer	*Via Nizza 45*	10100 Torino	Tel. 011-214235
Radio Tv Mirafiori	*C.so Un. Sovietica 381*	10135 Torino	Tel. 011-616190/ 6197189
SMT Elettronica	*Via Bibiana 83/B*	10147 Torino	Tel. 011-218243
Paul e Chico Videos.	*Via Vitt. Emanuele 52*	10023 Chieri (To)	Tel. 011-9470295
Bit Informatica	*Via V. Emanuele 154*	10073 Cirié (To)	Tel. 011-9205455
HI-FI Club	*C.so Francia 92C*	10093 Collegno (To)	Tel. 011-4110256
ICS	*Stradale Torino 73*	10015 Ivrea (To)	Tel. 0125-251797
Cerrutti Mauro	*C.so Torino 234*	10064 Pinerolo (To)	Tel. 0121-70316
Eurex	*C.so Indipendenza 5*	10086 Rivarolo C.se (To)	Tel. 0124-27984
Ditta Elettrogamma	*C.so Bormida*	13100 Vercelli	Tel. 0161-53689
Elettronica	*Strada Torino 15*	13100 Vercelli	Tel. 0161-393163
C.S.I. Teorema	*Via Losana 9*	13051 Biella (Vc)	Tel. 015-28622
Fotostudio Trevisan	*Via XXV Aprile 24/B*	13014 Cossato (VC)	Tel. 015-921431
Studio Fotogr. Imarisio	*P.zza Martiri Liberta' 7*	13039 Trino (Vc)	Tel. 0161-82081
F.lli Gatti	*Via Festaz 75*	11100 Aosta	Tel. 0165-35659

PRODOTTI SYSTEMS EDITORIALE

Software su cassetta

La voce III	L.12000
Raffaello	L.10000
Oroscopo	L.12000
Computer-Music	L.12000
Gestione familiare	L.12000
Banca dati	L.12000
Dichiarazione dei redditi (740/S)	L.16000
Matematica finanziaria	L.20000
Analisi di bilancio	L.20000
Arredare (richiede linguaggio Simon's Basic)	L.10000

Software su disco

Ms-Dos & Gw-Basic	L.25000
Ms-Dos & Gw-Basic (con prova di acquisto cassetta)	L.15000
La voce III	L.12000
Raffaello	L.10000
Oroscopo	L.12000
Computer-Music	L.12000
Gestione familiare	L.12000
Banca dati	L.12000
Dichiarazione dei redditi (740/S)	L.24000
Matematica finanziaria	L.20000
Analisi di bilancio	L.20000
Arredare (richiede linguaggio Simon's Basic)	L.20000
Graphic Expander C/128 in modo 80 colonne	L.27000
Linguaggio macchina + Routine grafiche	L.12000

Offerta "Commodore speciale L.M." + dischetto L.16000

Libri

64 programmi per il Commodore 64	L. 4800
I miei amici C/16 e Plus/4	L. 7000
Strategie vincenti per Commodore 64	L. 5800
62 programmi per Vic 20, C/16 e Plus/4	L. 6500
Utilities e giochi didattici	L. 6500
Tutti i segreti dello Spectrum	L. 7000
Simulazioni e test per la didattica	L. 7000
Impara giocando il Basic dello Spectrum	L. 7000
Micro Pascal per Commodore 64/128	L. 7000
Dal registratore al drive del C/64	L. 7000
Ada	L. 5000
Il linguaggio Pascal	L. 5000

Directory

Ciascun dischetto	L.12000

Arretrati

Ciascun numero arretrato di Commodore Computer Club	L. 5000
Ciascun numero arretrato di Personal Computer	L. 5000
Ciascun numero arretrato di VR Videoregistrare	L. 5000

Per un ottimale utilizzo del software "Matematica finanziaria" è opportuna la lettura degli articoli relativi pubblicati sui N.13, 14, 15 della rivista "Commodore" e sui N.1, 2 e 3 della Rivista Personal Computer.

Per un ottimale utilizzo del software "Analisi di Bilancio" è opportuna la lettura degli articoli relativi pubblicati sui N.2, 3, 5 della Rivista Personal Computer.

Per un ottimale utilizzo del software "Linguaggio Macchina e Routine grafiche per C/64" è opportuna la lettura del fascicolo "Commodore Speciale" appositamente dedicato.

Coloro che desiderano procurarsi i prodotti della Systems Editoriale devono inviare, oltre alla cifra risultante dalla somma dei singoli prodotti, la cifra di L.3000 per spese di imballo e spedizione, oppure L.6000 se si preferisce la spedizione per mezzo raccomandata.

Sconti e agevolazioni

Le spese di imballo e spedizione sono a carico della Systems se ciascun ordine è pari ad almeno L.50000 (di listino).

Gli abbonati hanno diritto allo sconto del 10% e alla spedizione gratuita se la somma totale raggiunge la cifra di L.50000 (di listino).

Oltre alla spedizione gratuita, viene praticato uno sconto del 10% (per gli abbonati è del 20%) se la cifra raggiunta per ciascun ordine raggiunge le L.100000 (di listino).

Abbonamenti

Commodore Computer Club (11 fascicoli)	L.45000
Personal Computer (11 fascicoli)	L.40000
Commodore Computer Club + Personal Computer (11 + 11 fascicoli)	L.70000
VR Videoregistrare (12 numeri)	L.45000

N.B.: la cifra per gli abbonamenti non può essere conteggiata per ottenere gli sconti e le agevolazioni di cui sopra.

Non è assolutamente possibile inviare materiale contrassegno.

Compilate un normale modulo di C/C postale indirizzando a:

C/C postale N. 37952207
Systems Editoriale
Viale Famagosta, 75
20142 Milano

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo di recapito telefonico, ma anche il materiale desiderato.

In ogni caso sarebbe opportuno inviare la presente scheda, debitamente compilata, allegando la fotocopia della ricevuta del versamento effettuato.

Chi volesse ricevere più celermente la confezione deve inviare la somma richiesta mediante assegno circolare oppure normale assegno bancario (non trasferibile o barrato due volte) intestato a: Systems Editoriale - Milano.